序

2001年7月间，电子工业出版社的领导同志邀请各高校十几位通信领域方面的老师，商量引进国外教材问题。与会同志对出版社提出的计划十分赞同，大家认为，这对我国通信事业、特别是对高等院校通信学科的教学工作会很有好处。

教材建设是高校教学建设的主要内容之一。编写、出版一本好的教材，意味着开设了一门好的课程，甚至可能预示着一个崭新学科的诞生。20世纪40年代MIT林肯实验室出版的一套28本雷达丛书，对近代电子学科、特别是对雷达技术的推动作用，就是一个很好的例子。

我国领导部门对教材建设一直非常重视。20世纪80年代，在原教委教材编审委员会的领导下，汇集了高等院校几百位富有教学经验的专家，编写、出版了一大批教材；很多院校还根据学校的特点和需要，陆续编写了大量的讲义和参考书。这些教材对高校的教学工作发挥了极好的作用。近年来，随着教学改革不断深入和科学技术的飞速进步，有的教材内容已比较陈旧、落后，难以适应教学的要求，特别是在电子学和通信技术发展神速、可以讲是日新月异的今天，如何适应这种情况，更是一个必须认真考虑的问题。解决这个问题，除了依靠高校的老师和专家撰写新的符合要求的教科书外，引进和出版一些国外优秀电子与通信教材，尤其是有选择地引进一批英文原版教材，是会有好处的。

一年多来，电子工业出版社为此做了很多工作。他们成立了一个“国外电子与通信教材系列”项目组，选派了富有经验的业务骨干负责有关工作，收集了230余种通信教材和参考书的详细资料，调来了100余种原版教材样书，依靠由20余位专家组成的出版委员会，从中精选了40多种，内容丰富，覆盖了电路理论与应用、信号与系统、数字信号处理、微电子、通信系统、电磁场与微波等方面，既可作为通信专业本科生和研究生的教学用书，也可作为有关专业人员的参考材料。此外，这批教材，有的翻译为中文，还有部分教材直接影印出版，以供教师用英语直接授课。希望这些教材的引进和出版对高校通信教学和教材改革能起一定作用。

在这里，我还要感谢参加工作的各位教授、专家、老师与参加翻译、编辑和出版的同志们。各位专家认真负责、严谨细致、不辞辛劳、不怕琐碎和精益求精的态度，充分体现了中国教育工作者和出版工作者的良好美德。

随着我国经济建设的发展和科学技术的不断进步，对高校教学工作会不断提出新的要求和希望。我想，无论如何，要做好引进国外教材的工作，一定要联系我国的实际。教材和学术专著不同，既要注意科学性、学术性，也要重视可读性，要深入浅出，便于读者自学；引进的教材要适应高校教学改革的需要，针对目前一些教材内容较为陈旧的问题，有目的地引进一些先进的和正在发展中的交叉学科的参考书；要与国内出版的教材相配套，安排好出版英文原版教材和翻译教材的比例。我们努力使这套教材能尽量满足上述要求，希望它们能放在学生们的课桌上，发挥一定的作用。

最后，预祝“国外电子与通信教材系列”项目取得成功，为我国电子与通信教学和通信产业的发展培土施肥。也恳切希望读者能对这些书籍的不足之处、特别是翻译中存在的问题，提出意见和建议，以便再版时更正。

吴佑寿
中国工程院院士、清华大学教授
“国外电子与通信教材系列”出版委员会主任

出 版 说 明

进入21世纪以来，我国信息产业在生产和科研方面都大大加快了发展速度，并已成为国民经济发展的支柱产业之一。但是，与世界上其他信息产业发达的国家相比，我国在技术开发、教育培训等方面都还存在着较大的差距。特别是在加入WTO后的今天，我国信息产业面临着国外竞争对手的严峻挑战。

作为我国信息产业的专业科技出版社，我们始终关注着全球电子信息技术的发展方向，始终把引进国外优秀电子与通信信息技术教材和专业书籍放在我们工作的重要位置上。在2000年至2001年间，我社先后从世界著名出版公司引进出版了40余种教材，形成了一套"国外计算机科学教材系列"，在全国高校以及科研部门中受到了欢迎和好评，得到了计算机领域的广大教师与科研工作者的充分肯定。

引进和出版一些国外优秀电子与通信教材，尤其是有选择地引进一批英文原版教材，将有助于我国信息产业培养具有国际竞争能力的技术人才，也将有助于我国国内在电子与通信教学工作中掌握和跟踪国际发展水平。根据国内信息产业的现状、教育部《关于"十五"期间普通高等教育教材建设与改革的意见》的指示精神以及高等院校老师们反映的各种意见，我们决定引进"国外电子与通信教材系列"，并随后开展了大量准备工作。此次引进的国外电子与通信教材均来自国际著名出版商，其中影印教材约占一半。教材内容涉及的学科方向包括电路理论与应用、信号与系统、数字信号处理、微电子、通信系统、电磁场与微波等，其中既有本科专业课程教材，也有研究生课程教材，以适应不同院系、不同专业、不同层次的师生对教材的需求，广大师生可自由选择和自由组合使用。我们还将与国外出版商一起，陆续推出一些教材的教学支持资料，为授课教师提供帮助。

此外，"国外电子与通信教材系列"的引进和出版工作得到了教育部高等教育司的大力支持和帮助，其中的部分引进教材已通过"教育部高等学校电子信息科学与工程类专业教学指导委员会"的审核，并得到教育部高等教育司的批准，纳入了"教育部高等教育司推荐——国外优秀信息科学与技术系列教学用书"。

为做好该系列教材的翻译工作，我们聘请了清华大学、北京大学、北京邮电大学、东南大学、西安交通大学、天津大学、西安电子科技大学、电子科技大学等著名高校的教授和骨干教师参与教材的翻译和审校工作。许多教授在国内电子与通信专业领域享有较高的声望，具有丰富的教学经验，他们的渊博学识从根本上保证了教材的翻译质量和专业学术方面的严格与准确。我们在此对他们的辛勤工作与贡献表示衷心的感谢。此外，对于编辑的选择，我们达到了专业对口；对于从英文原书中发现的错误，我们通过与作者联络、从网上下载勘误表等方式，逐一进行了修订；同时，我们对审校、排版、印制质量进行了严格把关。

今后，我们将进一步加强同各高校教师的密切关系，努力引进更多的国外优秀教材和教学参考书，为我国电子与通信教材达到世界先进水平而努力。由于我们对国内外电子与通信教育的发展仍存在一些认识上的不足，在选题、翻译、出版等方面的工作中还有许多需要改进的地方，恳请广大师生和读者提出批评及建议。

电子工业出版社

教材出版委员会

译 者 序

本书的作者Emmanuel C. Ifeachor教授是国际上信号处理方面的著名专家，他的主要研究领域包括信号处理与建模技术及其在多媒体、音频和生物医学中的应用。Ifeachor教授撰写过多部有关信号处理方面的著作，并且曾经在许多由IEE、IEEE组织的国际技术委员会中任职。

本书以多年来为英国普利茅斯大学和Sheffield Hallam大学讲授数字信号处理方面的实践性本科课程以及给工业应用部门的应用工程师讲授的课程为基础，由作者精心编写而成。书中介绍了数字信号处理的基本理论及其应用，并通过实际的例子来阐述一些关键的技术专题。有关数字信号处理方面的教材有很多种，但这些教材普遍偏重理论的介绍，而较少涉及现实生活中的应用实例。这会使学生认为数字信号处理理论的数学公式太多，因此感到抽象、难学；另一方面，学生在学完这门课程之后，仍然不是很清楚DSP究竟能做什么，以及如何设计一个实际的系统。本书试图在理论和实践之间架起一座桥梁，并在介绍这些基础理论时尽可能减少数学推导。书中在应用方面的实例涉及无线电通信、数字音频、生物医学等许多与我们日常生活密切相关的技术领域，相信通过这些实例可以提高读者的学习兴趣，使读者掌握DSP的相关技术，并引导读者如何使用DSP来设计实际的工程系统。

本书的主要特点是：

- 用实际的例子和现实生活中的应用来阐述DSP技术
- 将数学知识减少到理解本书内容所需掌握的程度
- 提供了许多DSP算法的C语言和MATLAB实现
- 讲解了实时DSP系统的模拟I/O接口
- 介绍了新型DSP处理器的结构和硬件
- 包括了系统识别、反卷积、小波变换和参数谱分析等新课题
- 提供了设计DSP系统的指导和实例
- 相关的应用领域包括音频、生物医学、无线电通信等

参加本书翻译工作的有罗鹏飞（第1章~第4章，第7章）、杨世海（第9章~第11章，第13章）、朱国富（第12章，第14章）和谭全元（第5章~第6章，第8章）。此外，赵艳丽、熊跃军、罗佳莹、张文明、刘忠、杨建华、王世希、来庆福、陈英、江晶、曾勇府、彭岁阳、张剑、肖旭、何志华、王象、谢小霞、丹梅、徐振海也对本书的翻译、校对和资料整理工作提供了很多帮助。最后，由罗鹏飞对全书的译文进行了校对和整理。由于译者的水平有限，文中难免有不妥之处，敬请读者不吝赐教。

前　言

本书的写作目的

在过去的几年中，数字信号处理（DSP）在许多关键性的技术领域继续产生着重要的影响，并且这种影响正在日益增加。这些技术领域包括无线电通信、数字电视和媒体、生物医学、数字音频和仪器等。在许多新的和正在涌现的数字产品以及信息社会的许多应用（如数字蜂窝电话、数字相机和TV、数字音频系统等）中，DSP是其核心。自本书第一版出版以来，对于电子、计算机和通信工程师来说，精通DSP的愿望已经大大增强。现在，DSP是大多数电子/计算机/通信工程专业的课程的核心内容。

通过提供基于MATLAB的习题、相关的指导手册以及Web资源，本书的第二版经过了重新的整理和修改，并且包含了一些附加的主题，这些主题的重要性正在日益增长。增加这些内容的目的是为了适应软件开发和信息技术的更为广泛的适用性、信号处理教学的发展以及读者的要求。在大学的教学和科研活动中，对基于Web的资源和MATLAB这样的信号处理软件工具的使用正在逐渐普及。因此，我们满足了读者对基于MATLAB的资源的需求。MATLAB使信号处理变得简单化，几个命令就可以立即显示结果。在开发信号处理算法和解题的过程中，我们也能从中获得乐趣，而不必专注于编写详细的程序。我们相信，本书的MATLAB例子和习题将增加学生的实践经验，而教师可以获得更多可用的教学资源。

正如本书的第一版，第二版的目标是在理论与实践之间架起一座桥梁。因此，我们继续保持本书的主要特征，即覆盖现代DSP的主要课题，并且提供实际的例子与应用。与第一版一样，我们将实际的例子和系统与理论相结合，以保持学生的学习兴趣，增强学习的动力。第二版的许多章节都经过了大量的修改，增加了最新的信息，并尽量使内容更加简洁。我们对每章末尾的习题加以扩展，以便检验、巩固和加深学生对内容的理解。在修订本书的时候，加入了自第一版出版后我们在DSP方面取得的经验，并吸取了世界各地读者的反馈意见。

新引入的主题包括：在模/数转换中的过抽样和带通抽样技术，这些技术利用了DSP提供的优势；用于信号的时频表示和分辨的小波变换；从未知系统的输出识别输入信号的盲解卷积；可用于短信号高分辨率的参数谱估计；新的DSP处理器的结构以及在定点DSP系统中降低舍入噪声的实际方法；用于辅助复习的基于计算机的多项选择题。在本书中提供了基于MATLAB的例子和习题。

本书以多年来为英国普利茅斯大学和Sheffield Hallam大学讲授的数字信号处理方面的实践性本科课程以及给工业应用部门的应用工程师讲授的课程为基础，由作者精心编写而成。我们觉得现在的许多教材对于本科生来说内容过于基本，或者对于相关领域的应用工程师来说又过于理论化。大多数的读者都有过这种经历，学习任何一个学科的基础知识与实际应用的差距是很大的。因此，我们决定撰写这本教材，相信本科生可以理解并赏识这本书，而且可以承担实际的数字信号处理课题；我们也相信，硕士生、博士生以及工程师也会发现本书是相当有价值的。

我们在应用DSP方面二十多年的研究工作对撰写本书的内容很有帮助。我们从这些研究工作中总结出了一些可供讨论的实际问题，为理论概念和工程实现架起桥梁，并且还提供了一些应用实例、案例研究和相关的习题。

DSP在工业界和大学方面的巨大影响力与发展仍将持续。众多实用的数字信号处理器证明了DSP在商业上的重要性。DSP的主要吸引力在于它具有达到要求的精度和出色的可再生能力，以及与模拟信号处理相比固有的灵活性。在工业界的许多工程师仍然缺乏DSP方面的必要知识和专门技术，因而不能完全利用目前市场上功能强大的数字信号处理器的巨大潜能。为了使工程师能够利用这些数字信号处理器来设计实际的DSP系统，本书提供了必要的基础知识和实际指导。

在大学里，DSP通常被认为是电子工程专业的课程中数学主题较多的一门。根据我们的教学经验，本书精简了有关的数学概念，保留了有用的、基本的和能够引起学习兴趣的内容，同时也强调了其中的一些难点。经验表明，学生如果能够意识到理论与实践的结合会学得更好，而掌握更多的理论内容对于知识的完备性也是必需的，我们对精通实际的知识和技能的学生充满信心。作者正是基于以上考虑编写本书的。

本书并未包含DSP的全部内容，而是覆盖了电气、电子和通信工程等专业的课程的多个方面，同时也涵盖了许多与工业界具有某些特定关系的DSP技术。在过去的几年里，我们开始将这些技术（包括自适应滤波和多速率处理）融入到本科生的教学中。

本书强调DSP的实际应用方面。第二版的一个很重要的特征是包括了MATLAB的例子，以及信号处理、分析、设计与考察时间-效率方法等方面的习题。鼓励读者运行MATLAB程序来加深对DSP的理解。我们也提供了经过修改的第一版中的C语言DSP软件工具，实践证明这些工具是很流行的。

MATLAB在工业界和大学中作为基本工具而得到了广泛使用，它要求的编程技能比C语言少。MATLAB带有良好的图形和显示工具，从而为DSP的开发提供了一个好的环境。我们相信MATLAB是学生熟悉和胜任今后工作的有用工具。本书所有的MATLAB m文件都可以通过Web获得电子文档，其中有一些MATLAB m文件用于实现第一版中的C语言程序所执行的类似任务。此外，在指导手册*A Practical Guide for MATLAB and C Language Implementations of DSP Algorithms*的CD上也包含了m文件（以及第一版的C语言程序），详细情况请参见后面的内容。

本书的主要特征

- 从实践的观点出发，提供了DSP技术的基础、实现和应用的理解。
- 清晰性和易于阅读，将数学知识减少到对于理解本书内容所需掌握的程度。
- DSP的技术和概念通过一些实践化和处理过的实例加以说明，从而加深对DSP技术的理解。
- 为读者能够设计和开发实际的DSP系统提供指导，我们给出了完成一个设计例子和实现的详细细节，其中包括DSP处理器的汇编语言。
- 给出一些能够体现实践经验的MATLAB例子和习题。
- 提供许多DSP算法和函数的C语言实现，这些程序包括：

  -数字FIR和IIR滤波器设计
  -用户设计的定点IIR滤波器有限字长效应分析
  -将串行实现结构转化为并行实现结构
  -相关计算
  -离散和快速傅里叶变换
  -z反变换
  -频率响应估计
  -多抽样率处理系统的设计

- 在Web上提供了基于PC的MATLAB m文件的电子文档（在指导手册的CD上也包含了第一版的C语言程序，详细情况请参见下面的内容）。
- 在每章的最后包含了许多习题，并且提供了用于复习的多项选择题。
- 使用了一些实际中的例子来说明一些重要的概念，从而加深对知识的理解。

本书的读者对象

本书的读者对象包括工程、科学、计算机科学等专业的学生，以及希望获得DSP方面的相关知识的应用工程师和技术人员。特别是电子、电气和通信工程等专业的毕业生，将会发现本书无论是作为教材还是辅助课题设计都是相当有价值的，学生在DSP方面的课题设计所占的课题比例有了很大的增长。而且，本书对以上专业攻读硕士或博士学位的研究生也是相当有帮助的。

大学本科生将会发现本书的基本主题是非常具有吸引力的；并且我们相信，无论是在他们的课程学习期间，还是对于今后进入到工业界从事研究工作，本书都是非常有价值的信息源。

许多商业和政府组织机构承担其内部DSP短期课程的培训，这些短期课程都是根据某本教材开设的。我们相信，本书可以作为一本很好的教材，同时也可以作为本科生、研究生和应用工程师自学的参考书。

本书的组织结构

第1章包括了DSP的概述及其应用，使读者认识到DSP的含义及其重要性。第2章根据一些现实中的例子从实际的观点阐述了许多基本的主题，这些基本内容形成了DSP的基础，如信号的抽样、量化以及它们在实时DSP中的内在意义。同时还包括了如AD/DA转换中的过抽样技术、带通信号的抽样以及均匀和非均匀量化等重要的主题。在本章还介绍了离散时间信号与系统的概念，这些概念在第4章进行了深入的讨论。

离散变换，特别是离散傅里叶变换（DFT）和快速傅里叶变换（FFT），为DSP以及时域和频域分析提供了重要的数学工具，第3章对它们进行了介绍，并且讨论了相关的应用。从傅里叶变换以及指数傅里叶级数到离散傅里叶变换的推导给出了逻辑上的证明，因为DFT并不要求覆盖离散傅里叶级数的概念，因此也就没必要增加本书的内容，相关的讨论也限制在变换的描述和实现上。特别是在第3章里并没有涵盖加窗的知识，我们认为在第11章的谱分析中讨论加窗会更加合适。作为离散余弦变换的重要应用，在图像处理的JPEG标准中进行了描述。由于对非平稳信号的适用性以及在时域和频域分辨信号的能力，小波变换在许多领域的应用正在日益增加，因此在第3章中也包含了有关内容的介绍。这一章还描述了离散变换应用于信号去噪的多分辨分析和奇异检测。

在第4章中讨论了离散信号与系统的基础概念，描述了z变换方法，它是表示和分析离散信号与系统的非常重要的工具。本章重点分析了z变换的许多应用，例如在离散时间信号与系统中的频率响应的设计以及分析和计算应用的例子。在本书的其余部分，将通过一些实际的例子来说明z变换的概念及其应用。

相关和卷积是基本的并且是与DSP紧密相关的主题，我们在第5章中对其进行了深入的讲解。作者认为本章对于DSP是必不可少的，建议读者仔细学习本章的内容，逐步掌握相关的概念与方法。这些内容可能覆盖了几年的本科课程。在第二版中给出了一些附加的内容，如系统识别、解卷积和盲解卷积。盲解卷积特别有用，因为它利用了信息最大化，使得确定未知冲激响应系统输出端测得的未知输入信号成为可能。

第6章～第8章详细讨论了数字滤波器的设计实践，这是DSP很重要的内容之一，也是大多数DSP系统的核心。滤波器的设计是一个很大的主题，第6章提供了滤波器设计的一般框架，给出了数字滤波器设计的逐步指导。

从技术规范到实现的FIR滤波器设计技术在第7章进行了讨论。本章提供了几个可以执行的例子，以便加深对重要概念的理解。在这个新版本中，附加的一些内容包括频率FIR滤波器的自动设计，并给出了一个完整的设计实例，用来说明滤波器设计的各个阶段是如何综合在一起的。

第8章详细讨论了根据简单的逐步设计指导进行IIR（无限冲激响应）滤波器设计。本章已经经过了重新的组织和扩展，特别是为了清晰起见，重新安排了有关系数的计算，并根据读者的一些反馈信息加入了新的内容，以覆盖IIR滤波器设计的一些重要课题。此外，本章还给出了一些可以执行的例子，帮助读者理解从技术规范到实现的IIR滤波器设计过程。设计的例子用MATLAB和C语言给出。

我们已经对IIR滤波器设计进行了精简，将有限字长效应的内容移至第13章。我们采纳了读者的反馈意见，本书在第1章～第8章包含了大多数DSP课程的内容，更为先进的DSP知识将在以后的一些章节中出现。第13章将介绍DSP算法中的有限字长效应的处理。

多抽样率处理技术允许使用不止一个抽样率来处理数据，这样使得一位ADC和DAC（数/模转换）以及过抽样数字滤波等新技术可以使用，这些新技术应用于许多现代数字系统中，例如大家都熟悉的CD播放器。第9章通过一个处理过的例子和实际的多抽样率系统，介绍了多抽样率处理的基本知识。这一章的内容已经扩展到包含多相（polyphase）的概念。我们将许多设计的例子和应用综合在一起，从而说明多抽样率系统的原理与设计问题。

第10章介绍了自适应信号处理的常用算法：LMS（最小均方）算法和RLS（递归最小二乘）算法，这是自适应滤波的关键内容。本章只介绍一些必要的理论，主要是讲解实际的应用。

第11章介绍了在频域描述与研究信号的谱估计和分析这一重要内容。通过介绍参数谱估计软件包，我们对这种方法进行了详细的阐述。如果信号能够由正确阶数的模型精确地描述，那么参数谱估计可应用到短信号长度，与非参数谱估计方法相比能够提供高分辨率的谱估计。通过在脑电信号中诱发反应信号的自回归谱估计的应用实例，我们进一步分析了这种方法。对谱分析特别感兴趣的读者应该学习第11章和第3章，因为第11章强调解释，而在第3章给出了处理过的例子。掌握了这些内容的读者能够较好地胜任频域信号分析方面的工作。

在最近的十多年内，DSP硬件方面取得了巨大的发展，出现了许多实用的低成本数字信号处理器。为了在DSP中成功地应用这些处理器，掌握DSP硬件和软件的概念是十分必要的。第12章讨论了DSP的通用和专用处理器的一些关键问题，DSP算法对这些处理器硬件和软件结构的影响，以及DSP功能的有效运行对结构的要求。本章的内容突出了当今DSP的新技术，特别是我们讨论了新的DSP结构，如长指令字、超标量以及新的定点和浮点DSP处理器（包括德州仪器公司的定点处理器TMS320C54和TMS320C62、摩托罗拉公司的定点处理器DSP56300、模拟器件公司的TigerSHARC IS0001）。

第13章详细讨论了在现代定点DSP系统中有限字长效应的分析，在适合采用定点精确算法的内容中提供了减少有限字长效应的解决方法。

第14章是全新的（尽管保留了第一版的一些内容）一章，可作为教师和学生进行教学和学习的资源。这一章包括了用于DSP算法实现的低成本DSP板的描述，以及现实生活中的应用实例，并通过情景学习的形式加以描述。其他的特点包括给出了基于计算机的多项选择题，这些问题覆盖了前面各章的一些关键概念，对于本书内容的复习是很有价值的。本章还描述了完整的实验室练习题，并且提供了情景学习和课题研究的思路。

如何使用本书

本科生教学的实用方法是通过第1章、第2章的内容来掌握一些基本概念（如抽样定理、离散信号与系统），了解DSP的应用及其优势。离散变换从第3章的DFT和FFT、第4章的z变换开始讲解，通过第11章和第5章来介绍DFT和FFT的应用。在第5章讨论了相关处理以后，应该详细分析数字滤波器。

根据我们的经验，给学生布置一定量的实践性作业，将使他们学到更多的知识。因此，我们鼓励学生完成这方面的一些课题，如滤波器的设计、z反变换、DFT和FFT。实验室也应该设计一些实验内容来进行演示，从而加深学生对所学内容的理解。我们认为，课堂教学与课外实践是同样重要的。

这些方法对本科毕业生和研究生的学习都是同样适用的，但是相应的进度应该更快一些，并且应该包括更多的有关多抽样率和自适应滤波器方面的专题学习。

本书的相关网站及配套的CD和指导手册

有关本书的附加信息在如下站点中可以找到：

www.booksites.net/ifeachor

作者希望读者在上面的网页中通过"Contact us"按钮来反馈信息。所有MATLAB m文件的电子副本都可以从

www.booksites.net/ifeachor

上下载，这些电子副本包括许多MATLAB m文件，这些文件可以用来执行类似于第一版中用C语言实现的任务。MATLAB m文件、C语言程序和汇编语言代码可以在与指导手册配套的CD（与指导手册一起提供）上找到。取自第一版的C语言程序（进行了少量的修改）以可执行程序和源代码两种形式提供，如果要运行源代码而不是可执行代码则需要C编译器。这些程序是使用Borland Turbo C 2.0版的标准ANSI C编写的。由Pearson出版的指导手册 *A Practical Guide for MATLAB and C Language Implementations of DSP Algorithms* 中也包含了许多在本书中使用的MATLAB m文件和C语言程序的演示例子。

目　录

第1章 引　言

本章的目的是要解释数字信号处理（DSP）的含义及其益处，并且介绍DSP涉及到的基本DSP运算。为了使读者注意到DSP广泛的应用领域，我们还将讲解现实世界中的一些特定的例子，这些例子均来自与大多数读者相关的领域。

1.1 数字信号处理及其益处

所谓信号是指任何含有某种信息的变量，可以传送、显示或操作信号，典型的信号有

- 语音，例如我们在电话、无线电和日常生活中遇到的信号；
- 生物医学信号，例如脑电图（脑电信号）；
- 声音和音乐，例如激光唱盘复制的信号；
- 视频和图像，例如人们在电视上看到的信号；
- 雷达信号，用来确定远距离目标的距离和方位。

数字信号处理涉及到信号的数字描述，以及数字处理器应用于信号分析、修改和从信号中提取信息。自然界大多数的信号是模拟的，这也意味着它们是随时间连续变化的，代表了物理量的变化过程，例如声波。在DSP中应用得最流行的信号形式，则是通过模拟信号按有规则的时间间隔抽样并转换成数字形式而得到的。

处理数字信号的特殊理由可以是从信号中消除干扰或噪声，得到数据的频谱，或者把信号变换成更为适当的形式。DSP现在广泛应用于以前通过模拟方法实现的领域，或者应用于利用模拟方法难以实现或不可能实现的全新领域。DSP的吸引力在于它具有如下的显著优势：

- **保证精度**。精度只由所用的位数确定。
- **完美的再现性**。由于器件的容差不存在变化，部件之间可以得到相同的性能。例如，使用DSP技术，数字记录可以复制多次而信号质量没有任何衰减。
- 性能不随温度和时间漂移。
- 利用半导体技术方面的巨大进展所带来的优势可以达到更大的灵活性、更小的尺寸、更低的成本、低功耗和更高的速度。
- **更大的灵活性**。DSP系统可以进行编程或者再编程来执行许多功能而不需要修改硬件，这或许是DSP系统最重要的特征。
- **优良的性能**。DSP可以执行许多模拟系统不可能完成的性能，例如可以实现线性相位响应，复杂的自适应滤波算法可以利用DSP技术来实现。
- 在某些情况下，信息可能已经数字化，DSP是惟一可行的选择。

DSP并非没有劣势，然而这些劣势随着新技术的出现正在消失：

- **速度与成本**。DSP设计可能是昂贵的，特别是当涉及到大带宽信号的时候。当前，快速ADC/DAC（模数转换器/数模转换器）要么太昂贵，要么对于宽带DSP应用没有足够的分辨率。

目前只有专用IC能够用来处理兆赫范围的信号，这些专用IC是相当昂贵的。此外，大多数DSP器件的处理速度仍不够快，它们只能处理中等带宽的信号，带宽为100 MHz的信号仍然只能用模拟方法处理。然而，DSP器件的速度正在变得越来越快。

- **设计周期**。除非读者在DSP方面有着丰富的知识和必要的资源（软件包等），否则DSP设计是非常耗时的，某些情况下也几乎是不可能的，人们对该领域工程师的极度短缺有着广泛的共识。然而，这一现象随着许多大学生具有了数字技术的某些知识而正在发生变化，商业公司在他们的产品中正在利用DSP的优势。
- **有限字长问题**。在实时情况下，经济上的考虑常常意味着DSP算法只能采用有限位数来实现。在某些DSP系统中，如果没有采用足够的位数来表示变量，那么可能导致系统的性能恶化。

1.2 应用领域

DSP是现代电子技术增长最快的领域，在许多以数字形式处理信息或用数字处理器控制的领域得到了广泛的应用，这些应用领域包括：

- **图像处理**
  - 模式识别
  - 机器人视觉
  - 图像增强
  - 传真
  - 卫星气象图
  - 动画
- **仪表与控制**
  - 谱分析
  - 位置与速率控制
  - 噪声消除
  - 数据压缩
- **语音/音频**
  - 语音识别
  - 语音合成
  - 文本到语音
  - 数字音频
  - 均衡
- **军事**
  - 保密通信
  - 雷达处理
  - 声呐处理
  - 导弹制导
- **无线电通信**
  - 回波对消
  - 自适应均衡

  - –ADPCM 编码器
  - – 扩谱
  - – 视频会议
  - – 数据通信
- **生物医学**
  - – 病人监控
  - – 扫描仪
  - –EEG 脑电仪
  - –ECG 分析
  - –X 光存储 / 增强
- **消费应用**
  - – 数字、便携移动电话
  - – 多功能移动通信系统
  - – 数字电视
  - – 数字相机
  - – 网络电话、音乐和视频
  - – 数字应答机、传真和调制解调器
  - – 语音信箱
  - – 交互式娱乐系统
  - – 汽车的主动悬挂

参考上述并不完全的列表，我们就可以确信DSP的重要性。DSP重要性的最好体现是半导体制造商对DSP器件的频繁介绍。然而在该领域没有足够多的掌握DSP技术的工程师。本书的目标就是提供DSP技术的理解和实现，使读者能够获得这一重要学科的相关知识。

1.3 关键的DSP运算

现在已存在许多DSP算法，并且人们正在发明或发现更多的DSP算法。然而所有这些算法，包括大多数复数形式的算法都要求一些类似的运算，在一开始考察一下这些运算是有益的，分析这些算法有利于简化DSP的实现。基本的DSP运算是卷积、相关、滤波、变换和调制，表1.1总结了这些运算，下面给出每种运算的简单描述。表中需要注意的重要一点是所有基本的DSP运算都只要求乘、加 / 减和移位等简单的算术运算，同时要注意运算之间的相似性。

1.3.1 卷积

卷积是DSP使用最为频繁的一种运算。例如，它是数字滤波中的一种基本运算。给定两个有限长度的因果序列$x(n)$和$h(n)$，长度分别为N_1和N_2，它们的卷积定义为

$$y(n) = h(n) \circledast x(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k) = \sum_{k=0}^{\infty} h(k)x(n-k), \quad n = 0, 1, \ldots, (M-1)$$

其中符号$\circledast$用来表示卷积，$M = N_1+N_2-1$，在后面几章我们将会看到，DSP器件制造商已经开发的信号处理器可以有效地执行卷积运算中所包含的乘–累加运算。图1.1(c)给出了图1.1(a)和图1.1(b)

描述的两个序列卷积的例子。在这个例子中，$h(n)$（$n = 0, 1, 2, \ldots,$）可以看作为数字系统的冲激响应，$y(n)$是系统对输入序列$x(n)$的响应，卷积的数值即$y(n)$可以直接计算1.1式而得到。例如，$y(1)$可以求得如下：

$$y(1) = h(0)x(1) + h(1)x(0) + h(2)x(-1) + \ldots + h(12)x(-11)$$

$$= 0 \times 1 + (-0.02) \times 1 + 0 \times 0 + \ldots + 0 \times 0$$

$$= -0.02$$

当我们在频域进行考察时，卷积的重要性是十分明显的，其中利用了这样一个事实，即时域卷积等价于频域相乘。第5章给出了卷积的详细讨论，包括它的性质和图形解释。

表1.1　关键的DSP运算总结

(1) 卷积　给定两个有限长度序列$x(k)$和$h(k)$，长度分别为N_1和N_2，它们的线性卷积定义为

$$y(n) = h(n) \circledast x(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k) = \sum_{k=0}^{M-1} h(k)x(n-k),\ n = 0, 1, \ldots, M-1 \tag{1.1}$$

其中$M = N_1 + N_2 - 1$。

(2) 相关

(a) 给定两个长度为N的零均值序列$x(k)$和$y(k)$，它们的互相关的估计为

$$\rho_{xy}(n) = \frac{r_{xy}(n)}{[r_{xx}(0)r_{yy}(0)]^{1/2}} \quad n = 0, \pm 1, \pm 2, \ldots \tag{1.2}$$

其中$r_{xy}(n)$是互协方差的估计，它的定义为

$$r_{xy}(n) = \begin{cases} \dfrac{1}{N}\displaystyle\sum_{k=0}^{N-n-1} x(k)y(k+n) & n = 0, 1, 2, \ldots \\ \dfrac{1}{N}\displaystyle\sum_{k=0}^{N+n-1} x(k-n)y(k) & n = 0, -1, -2, \ldots \end{cases}$$

$$r_{xx}(0) = \frac{1}{N}\sum_{k=0}^{N-1}[x(k)]^2,\ r_{yy}(0) = \frac{1}{N}\sum_{k=0}^{N-1}[y(k)]^2$$

(b) 均值为零、长度为N的序列$x(k)$的自相关$\rho_{xx}(n)$的估计为

$$\rho_{xx}(n) = \frac{r_{xx}(n)}{r_{xx}(0)} \quad n = 0, \pm 1, \pm 2, \ldots \tag{1.3}$$

其中$r_{xx}(n)$是自协方差的估计，它的定义为

$$r_{xx}(n) = \frac{1}{N}\sum_{k=0}^{N-n-1} x(k)x(k+n) \quad n = 0, 1, 2, \ldots$$

(3) 滤波　有限冲激响应（FIR）滤波的方程为

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k) \tag{1.4}$$

其中$x(k)$和$y(k)$分别是滤波器的输入和输出，$h(k)$（$k = 0, 1, \ldots, N-1$）是滤波器的系数。

(4) 离散变换

$$X(n) = \sum_{k=0}^{N-1} x(k)W^{kn},\ 其中\ W = \exp(-j2\pi/N) \tag{1.5}$$

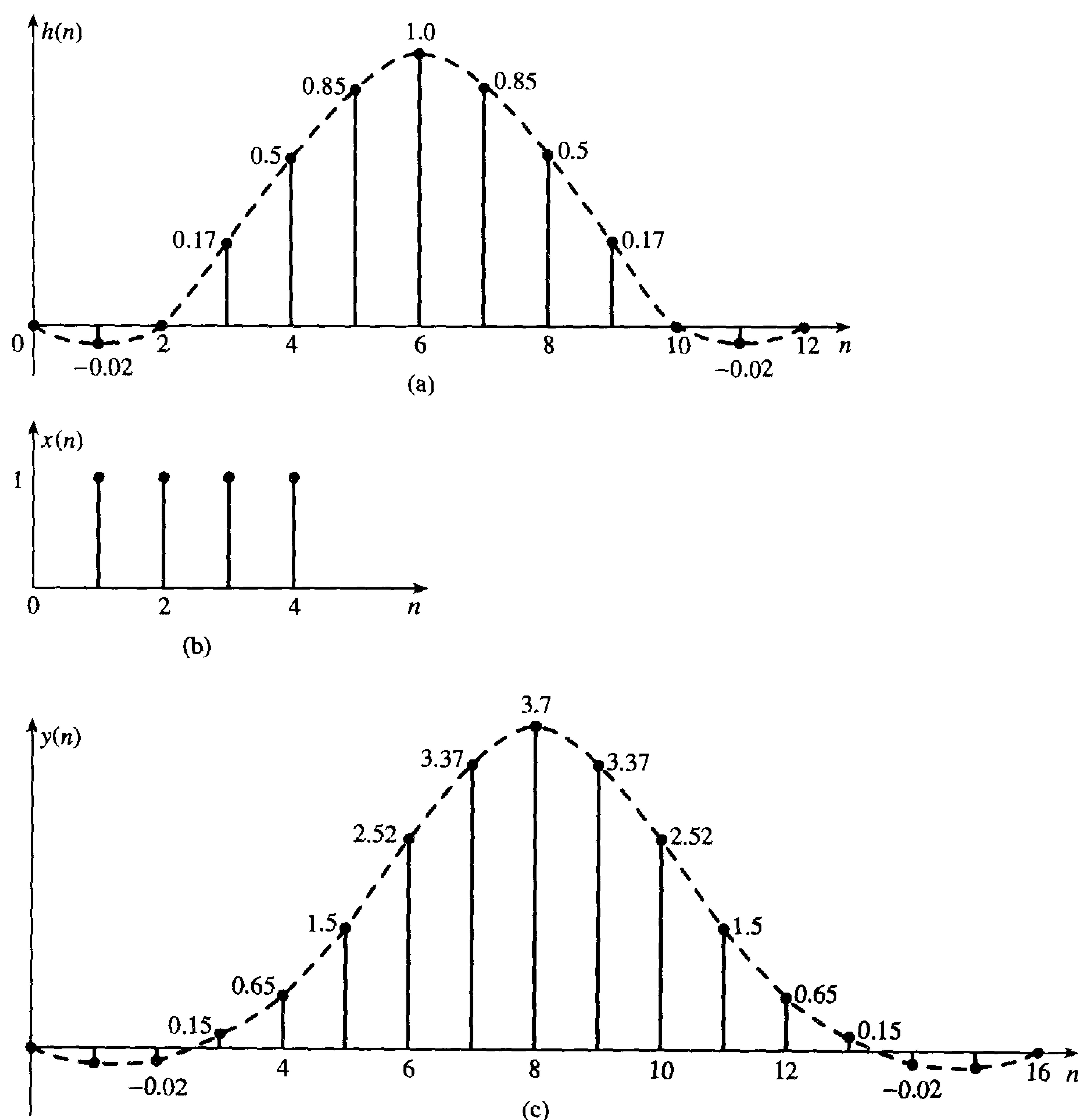


图 1.1 两个序列卷积的例子。$y(n)$是$h(n)$和$x(n)$的卷积，如果把$h(n)$看作为系统的冲激响应，那么$y(n)$是系统对输入$x(n)$的响应。以上$y(n)$的值直接由 1.1 式得到

1.3.2 相关

相关有两种形式：自相关和互相关。

(1) 互相关函数（CCF）是两个信号之间相似性或者共享性的量度。CCF的应用包括互谱分析、噪声中信号的检测/恢复，例如雷达回波信号的检测、模式匹配、延迟测量等。表 1.1 的 1.2 式给出了 CCF 的定义。

(2) 自相关函数（ACF）只涉及一个信号，并且提供了时域中信号结构或其行为的有关信息。ACF 是 CCF 的一种特殊形式，并且用于类似的应用。ACF 在检测隐藏的周期信号时有着特殊的用途。表 1.1 的 1.3 式给出了 ACF 的定义。

图 1.2 和图 1.3 给出了某个信号的 CCF 和 ACF。注意，受噪声污染的信号的 ACF 清楚地表明了噪声中存在一个周期信号（参见图 1.2）。而图 1.3 则说明了如何测量延迟。由系统引入的延迟量在 CCF 中是十分明显的，它可以通过从原点到最大峰值的时间来测得。

(a) 周期信号

幅度

时间（ms）

自相关

延迟（ms）

(b) 噪声

幅度

时间（ms）

自相关

延迟（ms）

(c) 周期信号加噪声

幅度

时间（ms）

自相关

延迟（ms）

图 1.2　自相关函数。注意，在(c)中噪声的周期信号，自相关函数的周期性依然是十分明显的，这也说明了为什么自相关可用来检测隐藏的周期信号

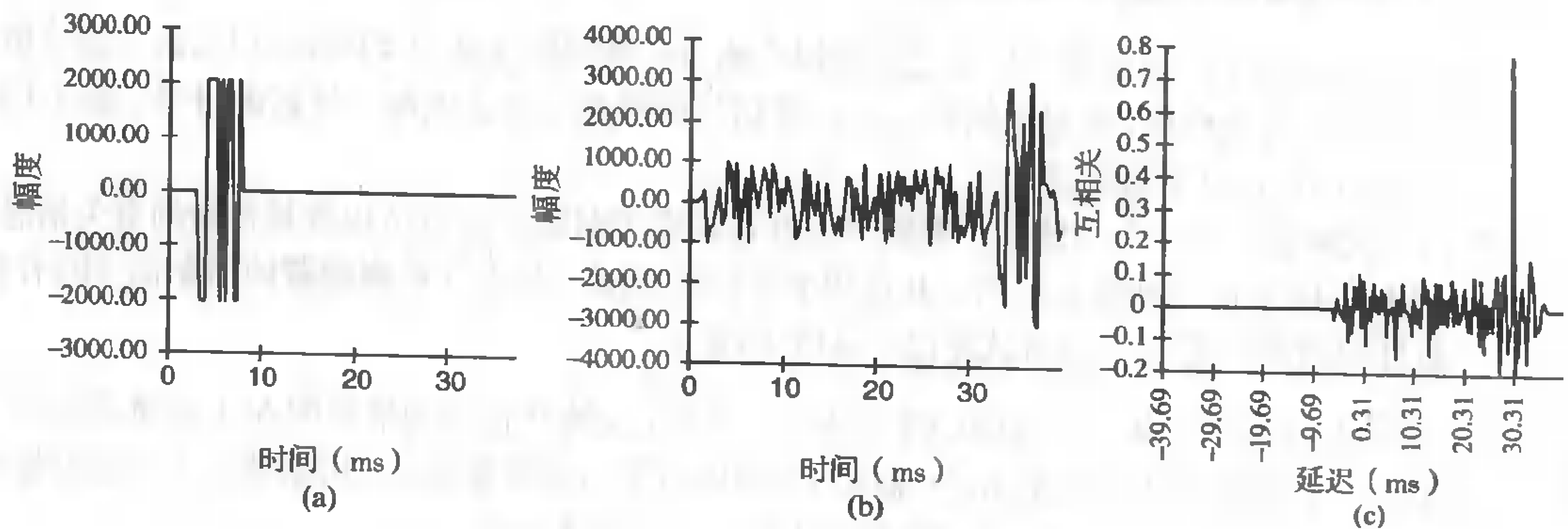


图 1.3　随机信号 $x(t)$和该信号延迟再叠加有噪声的信号 $y(t)$的互相关，两个信号的延迟刚好等于从原点到(c)图的其互相关峰值出现的时间

1.3.3 数字滤波

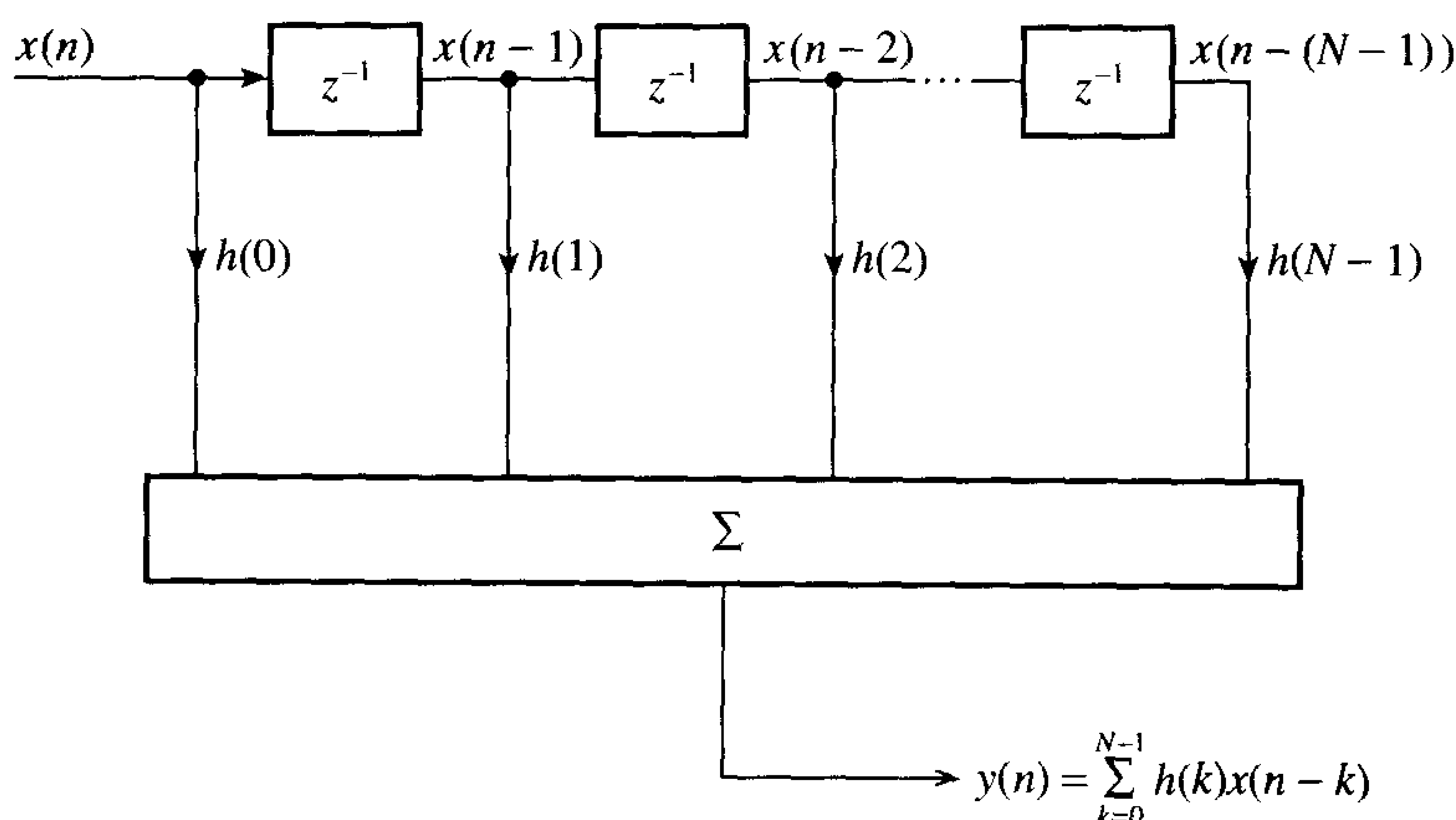
数字滤波是DSP中最重要的运算之一，重要的一类滤波器的数字滤波运算定义为

$$y(n)=\sum_{k=0}^{N-1}h(k)x(n-k)$$

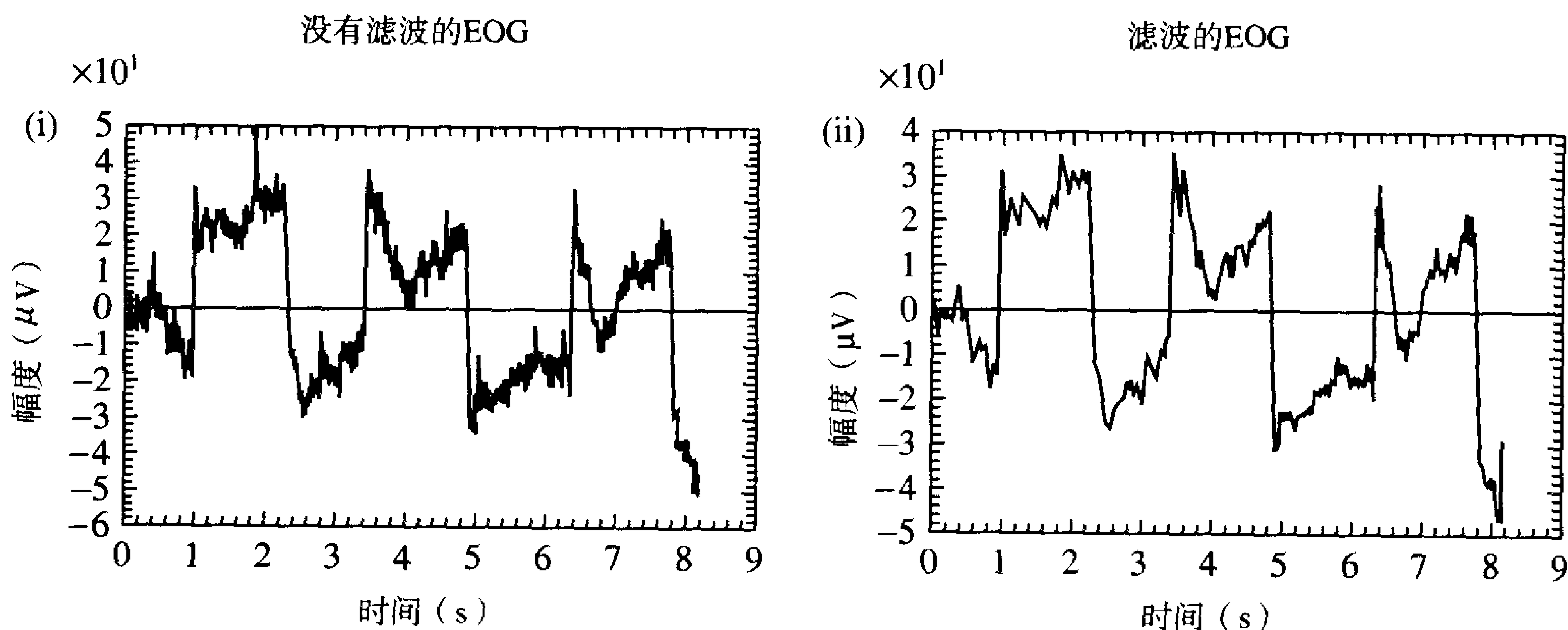
其中$h(k)$ ($k=0, 1,..., N-1$)是滤波器的系数，$x(n)$和$y(n)$分别是滤波器的输入和输出。对于给定的滤波器，它的系数是惟一的，这些系数确定了滤波器的特性。

我们注意到滤波实际上就是信号与滤波器的冲激响应$h(k)$在时域的卷积，图1.4(a)给出了以上定义的滤波器的框图。这种形式的滤波器通常称为横向滤波器，图中的z^{-1}表示一个抽样时间的延迟。

通常滤波的目的是为了从要求的信号中消除或者减少噪声，图1.4(b)表明了某个生物医学信号为了消去高频失真的数字低通滤波的效果。在这种应用中采用数字滤波器尤为重要，它可以使带内信号分量的失真达到最小。



(a) 横向滤波器的框图表示，$h(k)$ ($k=0, 1,..., N-1$) 是滤波器的系数，每个框中的z^{-1}表示一个抽样周期的延迟



(b) 对生物医学信号消去噪声的数字低通滤波

图1.4 1.3.3节定义的滤波器框图

1.3.4 离散变换

离散变换允许从频域来表示离散信号，或者允许离散信号在时域和频域之间进行变换。利用离散变换可以把信号分解成许多频率分量以得到信号的频谱。这种频谱的知识是非常有价值的，例如，我们可以确定发射信号要求的带宽。时域与频域之间的转换在许多DSP应用中也是十分必要的，例如，利用这种转换允许我们有效实现许多DSP算法，如数字滤波、卷积和相关。

人们开发了许多种离散变换，但离散傅里叶变换（DFT）是广泛采用的一种，它的定义为

$$X(k)=\sum_{n=0}^{N-1}x(n)W^{nk},\quad 其中\ W=\mathrm{e}^{-\mathrm{j}2\pi/N}$$

图 1.5 给出了 DFT 的一个例子，这里，利用 DFT 将滤波器的冲激响应 $h(n)$（$n=0,1,\ldots,N-1$）进行变换，得出滤波器的频率响应。DFT 及其应用的详细内容将在第 3 章、第 4 章和第 6 章给出。

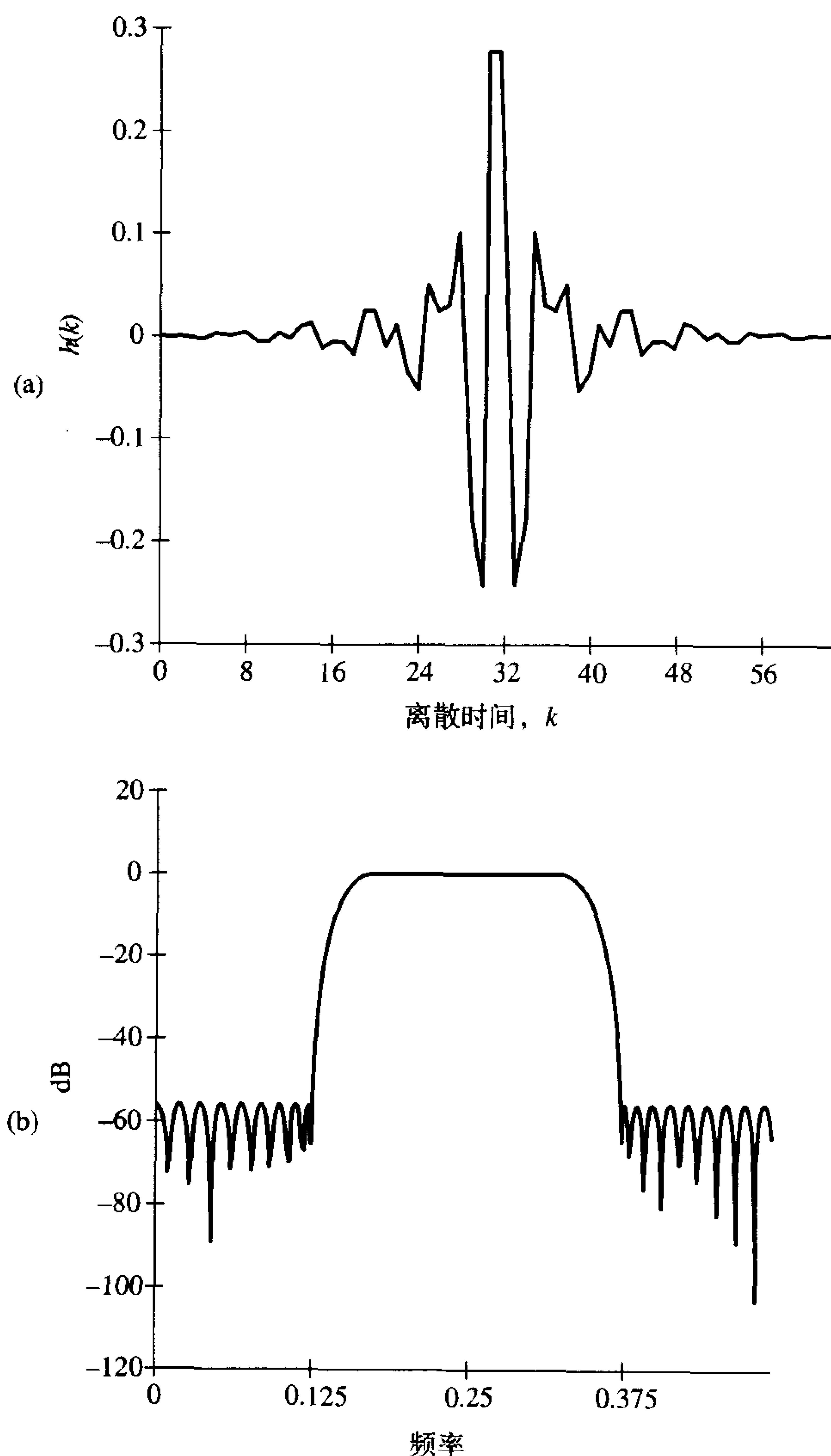


图 1.5 数字滤波器的时域和频域表示：(a)冲激响应；(b) 滤波器频谱。滤波器的频谱是由 $h(n)$的离散傅里叶变换得到的，这是 DFT 许多应用中的一个

1.3.5 调制

数字信号很少直接发射到较远的距离，或者以原始形式大量地保存。为了有效地利用可用传输系统的带宽，信号通常都经过调制，从而将它们的频率特性与传输系统的频率特性相匹配，或者与存储媒介相匹配以便使信号失真达到最小，或者确保信号具有某些期望的特性。两个广泛采用调制技术的领域是无线电通信和数字音频工程。

调制的过程常常包括改变高频信号的特性使其与我们希望发射的信号一致，这个信号称为调制信号，而高频信号称为载波。在带通信道（例如微波链）上发射数字数据的数字调制方法中有三个常用的调制方法，它们是幅移键控（ASK）、相移键控（PSK）和频移键控（FSK），当数字数据在全数字网络上发射的时候，通常采用脉冲编码调制（PCM）（Bellamy, 1982）。在数字音频中还采用了其他几种形式的调制方法，这些调制方法的详细内容可参看Watkinson (1987)的著作。

1.4 数字信号处理器

DSP系统是由实时运算来刻画的，并强调高吞吐率和大量的算术运算。最值得注意的运算是乘法、加法或乘－累加，这些运算导致大量的数据流通过处理器。

标准的微处理器结构不适合于DSP特征，这便导致了新一代处理器的开发，它的结构和指令集适合于DSP运算。新一代处理器或DSP芯片具有下列特征：

- 内置的硬件乘法器允许快速进行乘法运算，新的DSP芯片具有单周期乘－累加指令，其中一些会有几个乘法器并行工作。
- 程序和数据采用分离总线，分别存储，这就是著名的哈佛（Harvard）结构，它允许取指与执行重叠。
- 分支或循环的周期－保存指令，例如，下列德州仪器（Texas Instrument）的TMS320C25的数字滤波器的指令减少了周期数和程序大小：

```
RPTK N  ;重复下一条指令N次
MACD    ;将数据移入内存，相乘和延迟累加
```

- 非常快的速度，例如，TMS320C25使用40 MHz时钟，周期为100 ns。
- 采用流水线减少了指令时间，增加了速度。

更新的DSP芯片其速度更快、功能更强。现在的某些DSP芯片具有浮点运算能力，并且组合了标准微处理器的一些特征，如串行线、扩展存储空间、定时器和多级中断。DSP芯片的详细讨论以及如何使用DSP芯片进行设计将在第12章～第14章进行介绍。

1.5 DSP的实际应用概况

DSP技术是许多新的和正在涌现的支撑信息社会的数字信息产品和应用的核心，这些产品和应用要求收集、处理、分析、传输、显示和存储现实世界的信息，有时候则需要实时实现。DSP技术数字化地处理现实世界信号的能力，使得为大的消费市场（如数字蜂窝移动电话、数字电视和视频游戏）创造可生产的、新的、高质量的产品和应用成为可能。DSP在其他领域的影响力也是十分明显的，例如医学、健康监护（如病人监护、数字X光机、先进的心脏和脑电图系统），以及数字音

频（如CD播放器、音频混频器和电子音乐）和个人计算机系统（如有效数据存储的磁盘、误差校正、调制解调器、声卡和视频会议）。

DSP技术对我们的生活方式、工作方式和休闲方式的有益影响在图1.6中进行了说明。在后面三节中，我们将详细地描述一些新的应用，并着重强调DSP的应用方面。

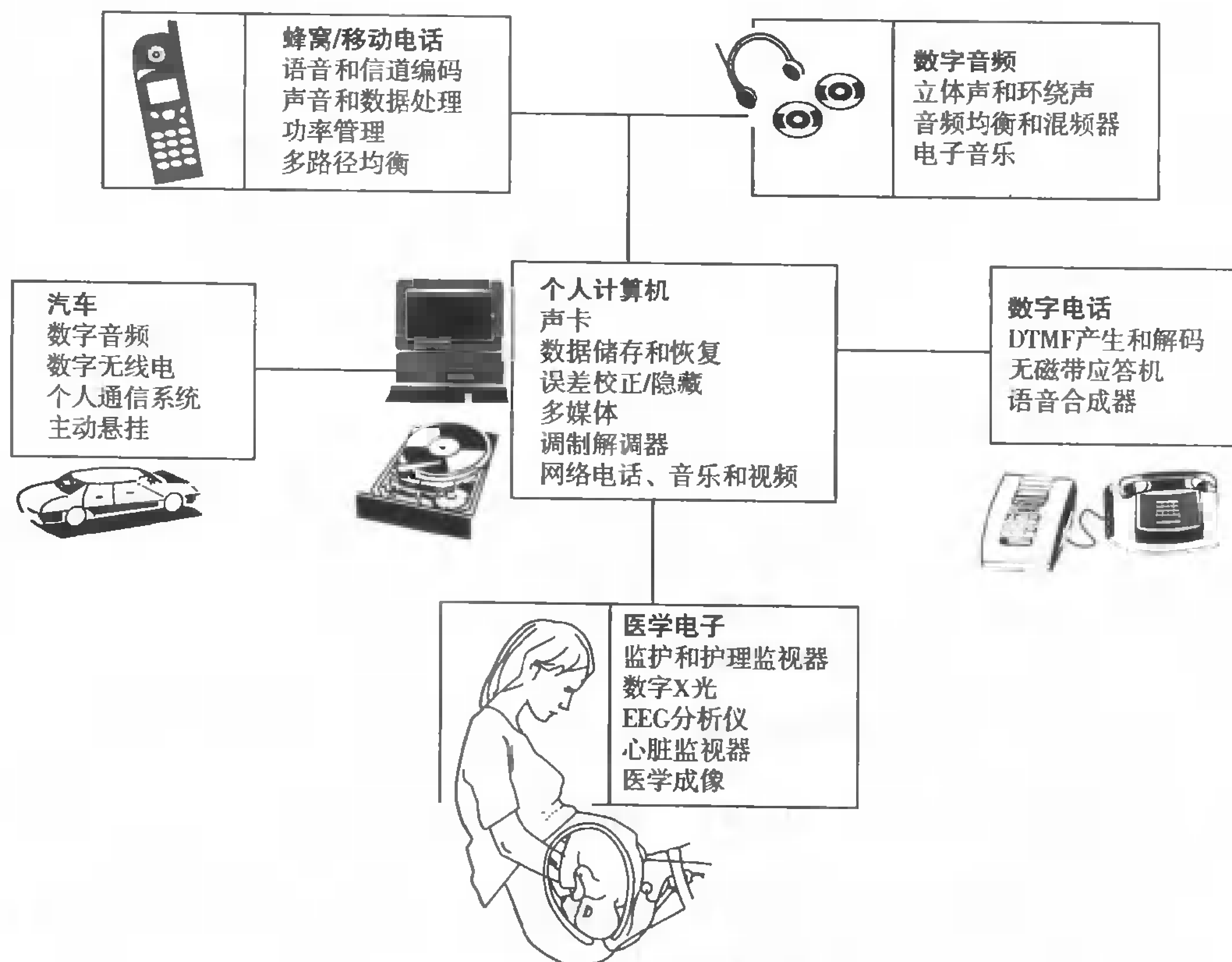


图1.6 DSP技术影响现代生活的实例

1.6 DSP的音频应用

1.6.1 数字音频混频

数字音频混频系统是DSP成功地用来改善音频质量和增强其功能的一个基本的例子。

音频混频应用于专业的和半专业的音频应用领域，例如录音棚、广播、声音增强、播音系统和直播。混频控制台允许对不同声源的多通道音频信号的特征进行调节、混频和监视，以满足特定应用的要求。

数字混频系统包括音频均衡、音频混频和后混频处理，请参见图1.7。数字混频均衡器（EQ）是一组具有可调特性的数字滤波器，用来控制音频输入的频带以便达到期望的声音（如推进或削减），类似于高音和低音控制。均衡的音频信号利用混频矩阵装置进行混频，混频矩阵装置允许任何一个及每一个音频输入混频到每一个输出信号。混频以后，进一步的信号处理运算包括混响与均衡。

混频系统具有交互式的控制来调节混频器的参数，如音量控制器（信号电平控制器）、实时EQ控制参数（滤波器的频率、Q和增益）。数字音频混频的挑战是以相当高的速率来达到混频参数的用户控制而没有声音失真（Clark et al., 2000）。每次用户调节控制面板的时候，混频器参数被修改以匹配新的要求，这样的调节可能导致声音的失真。在专业的音频混频系统中这是不能接受的。正如在Clark et al., (2000)中和第12章所讨论的那样，要达到专业的音频质量，混频算法的实现是关键。

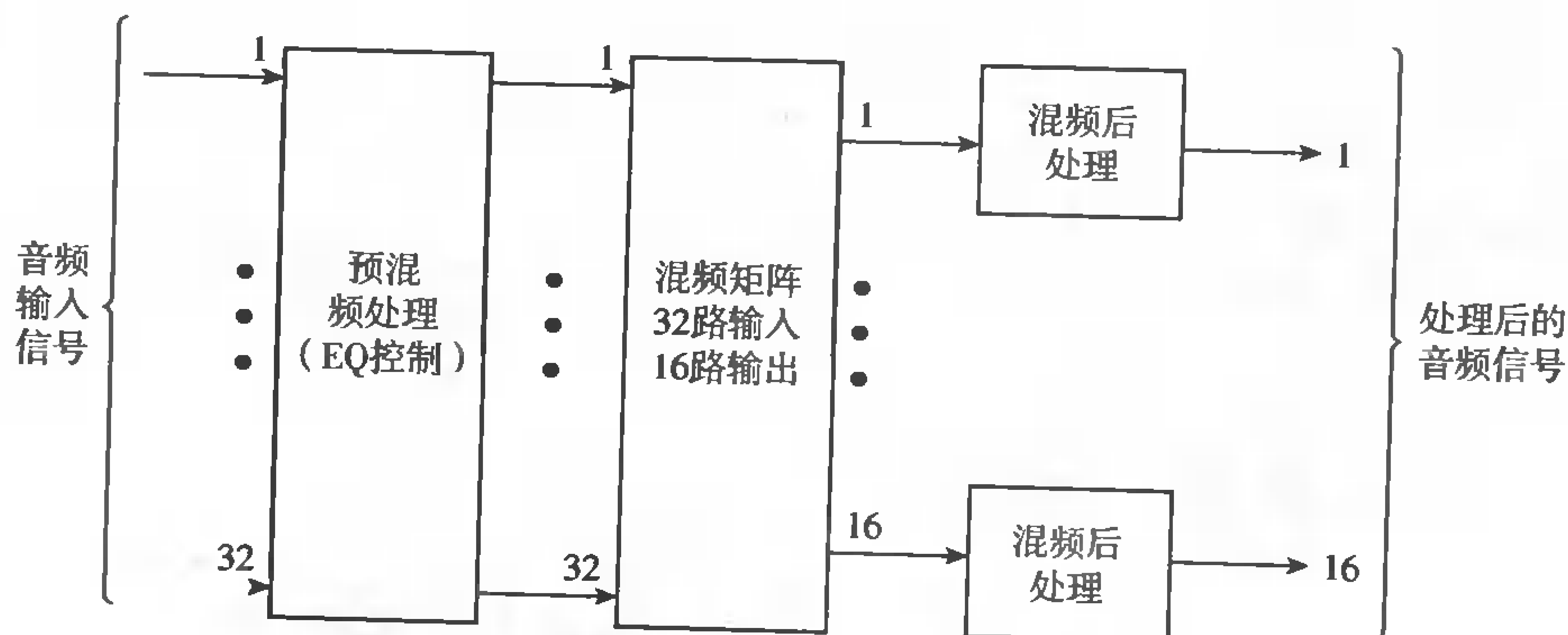


图1.7 32路输入、16路输出的立体声数字混频系统的简化框图

典型的混频器如图1.8所示，混频器的特征如下：

- 8路单声道输入（8个麦克风或8个视频输入）
- 两对立体声（左右）信号通道
- 中心控制的主编码条，如内存选择、功率放大设置和信号处理

图1.8描述的数字混频器使用了先进的DSP处理器来实现混频和混频后处理的音频信号处理新算法（如均衡、噪声选通、动态控制）。



图1.8 典型的8通道混频控制面板（Allen & Heath, Cornwall, UK）

1.6.2 语音合成和识别

1.6.2.1 语音合成

合成声音在过去是通过声音机械来感觉的。然而，半导体技术和DSP的进展，使得实现与人的语音没有差别的语音质量成为可能。

“Speak and Spell”（Frantz and Wiggins, 1982）是许多读者熟悉的语音输出的成功商业产品，它是小孩使用的电子学习辅助器，采用了LPC（线性预测编码）技术，实际上是将复制的语音看作为一个时变数字滤波器对一个周期的或随机的激励信号的响应（参见图1.9）。周期激励作为浊音（如元音），表示当声带振动时空气流通过声带；随机激励作为清辅音（如S、SH），表示通过声带的收缩强迫空气通过。滤波器模拟声带的行为，人的语音包含了许多冗余信息，LPC只保留反映语音特征的信息，如语调、口音、方言，并允许在中等大小的存储器中保持几分钟高质量的声音。

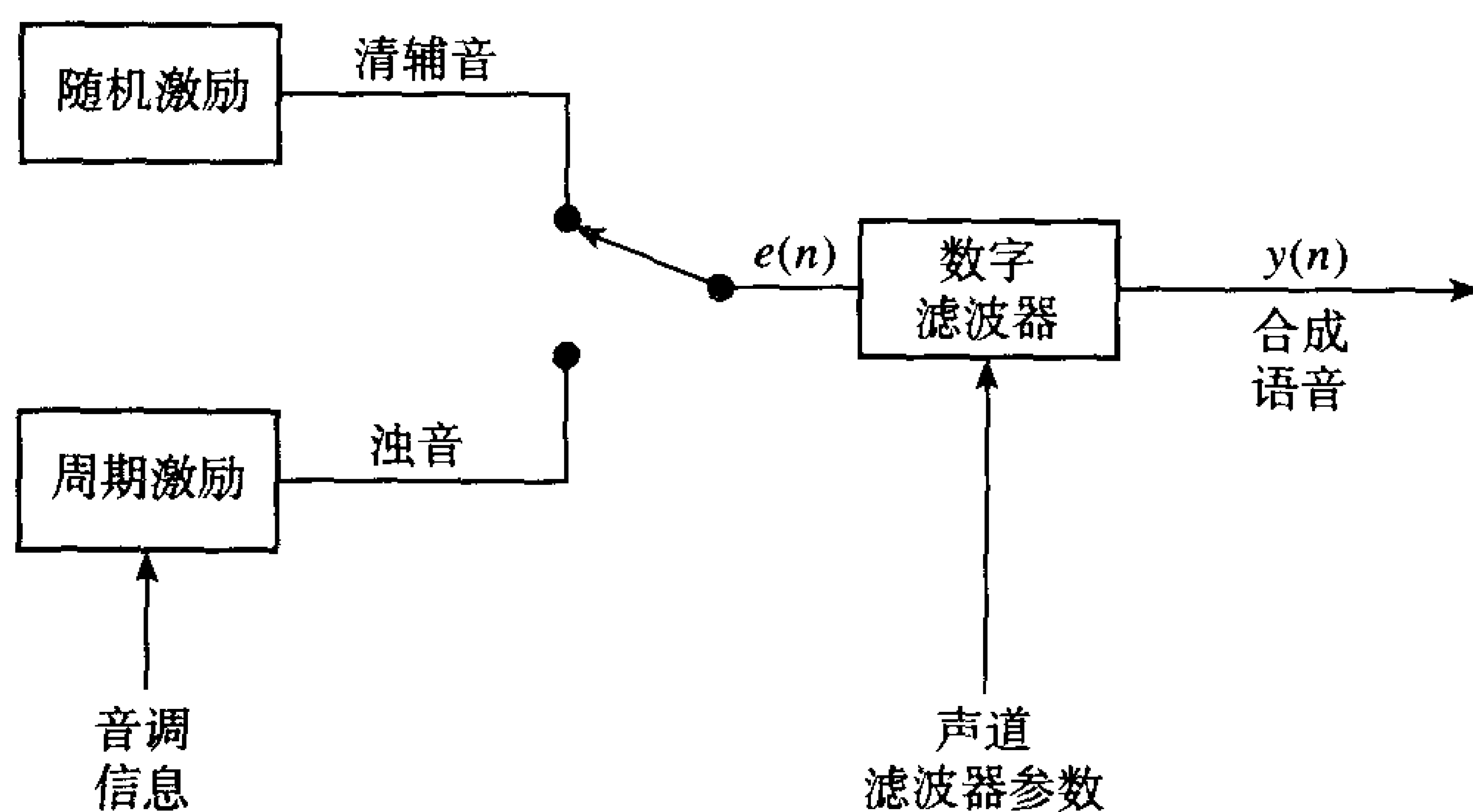


图1.9 语音的线性预测编码

在“Speak and Spell”中利用了TMS5100语音合成器芯片，这个芯片将LPC模型的所有分量以及译码器、8位DAC组合在一起。合成器芯片与4位微处理器、两个128 kb的ROM一起工作，128 kb的ROM可以保存300个单词和短语（参见图1.10）。语音信息以帧的形式存储在ROM中（代表25 ms的语音），每一帧由10个或12个LPC参数组成。为了更新数字滤波器的系数以及选择激励源和能量电平，帧的参数每25 ms加入到合成器一次，将数字滤波器的输出转换成模拟信号，并且应用到扬声器来产生具有特定音调、幅度和谐波内容要求的声音。为了每3 ms得到一次语音频谱的平稳转换，合成器通过对前一次和当前帧参数的内插来更新一次LPC参数。

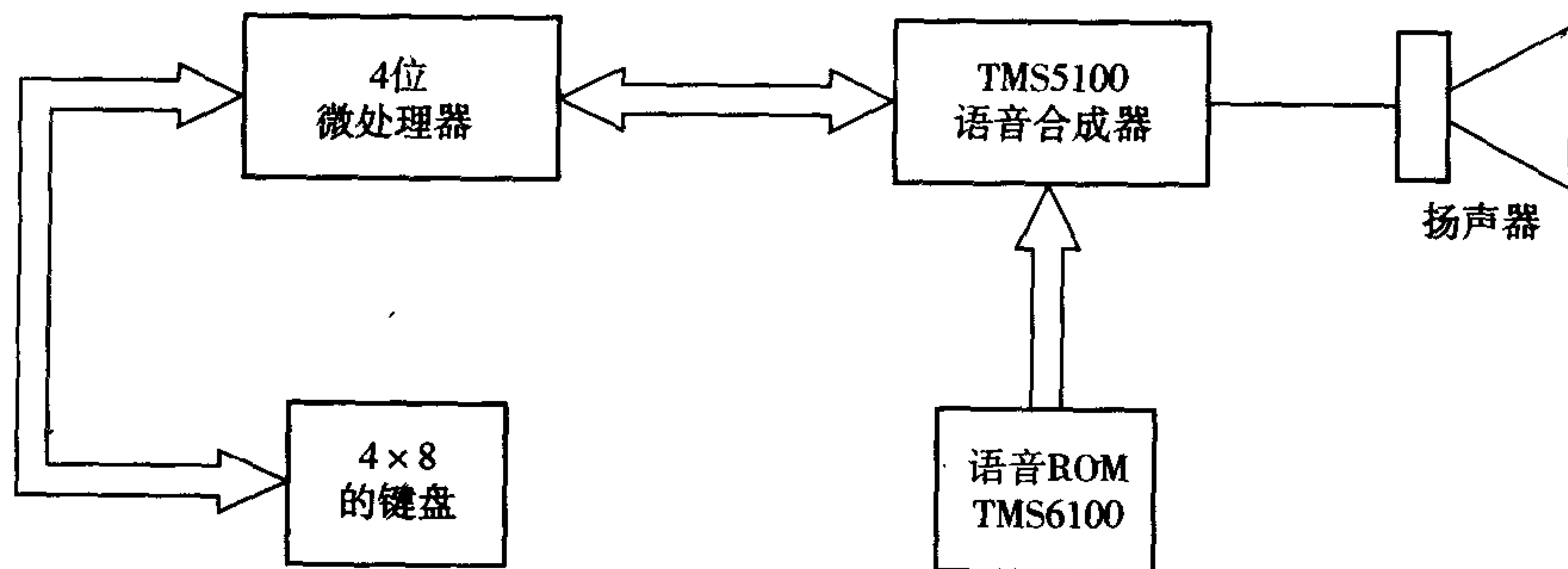


图1.10 “Speak and Spell”电子学习辅助器的结构

在一个操作模式下，要求小孩拼写一个单词。小孩输入一个单词，每次通过键盘输入一个字母，如果拼写是正确的，按“enter”，“Speak and Spell”响应“That is right”或者“Correct”；如果拼写不对，它会说“Wrong, try again”；如果下一次输入再次错了，它会责备地说“That is incorrect”，并加上“Correct spelling of ...is...”。

1.6.2.2 语音识别

声音识别包括将人的声音信息输入到计算机，计算机听取和识别人的语音。声音识别是一个正在积极研究的问题，它比语音合成要难得多。因此，成功的商业语音识别系统十分稀少，最成功的一个是“Speaker-Dependent Single-Word”系统，这个系统工作在两种模式下。在训练模式中，用户训练这个系统，通过说出每个待识别的单词并输入到麦克风来识别他或她的声音。系统将每个单词数字化，并且为每个单词创建一个模板存入它的存储器。当出现匹配时，就有一个单词被识别出来，系统通知用户或者采取某些行动。系统的性能受到说话者每个字停留的时间、背景噪声以及单词的发音清晰度的影响。在识别器中，两个重要的DSP运算是参数的提取和模式识别，参数提取运算将不同的模式从所说的单词中得到，并且用来创建一个模板，而模式的匹配是将模板与存储在存储器中的模板进行比较，如图1.11所示。

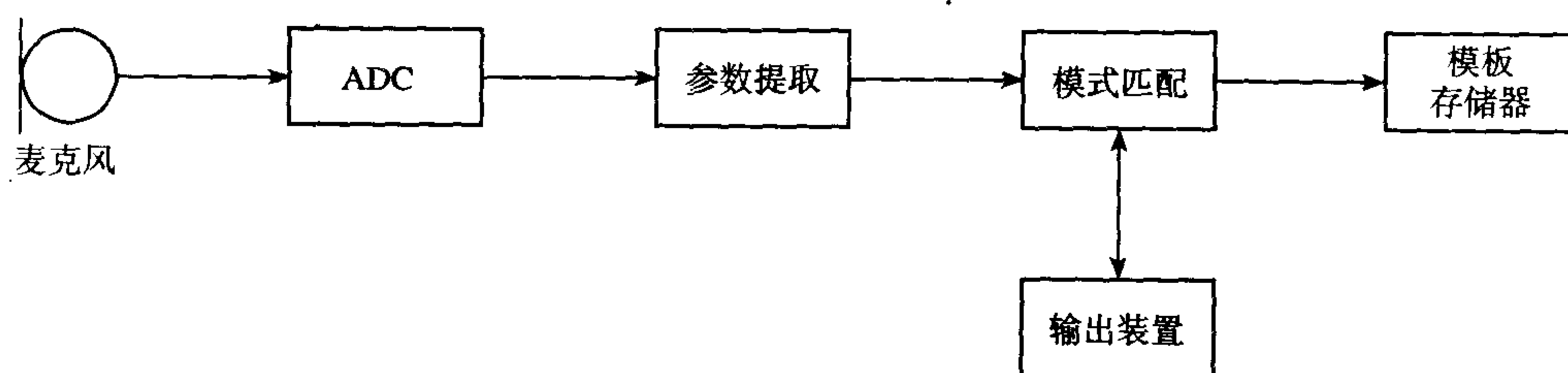


图1.11 语音识别系统的框图

对于大多数人来说，声音是最自然的一种通信形式。因此，在办公室环境里，声音系统允许应用程序由声音命令代替键盘输入。允许通常的办公室文件（如信件、备忘录）利用声音来创建和发送的系统正在受到重视，单词识别器正在加入到消费产品中，如用声音操作的电话拨号系统。单词识别器也用于为行动受限的残疾人提供的通过声音操作的产品中，这样就可以增加他们的生活独立性，使他们能够完成一些简单的任务，如打开/关闭电灯、收音机或电视机。

1.6.3 激光唱盘数字音频系统

大多数读者都熟悉在重放LP（long play）带的音乐时，如果存在损坏、刮伤、污迹或带子上有指印的情况，那么重放时会出现我们不愿意听到的声音。激光唱盘（CD）系统是一种先进的音频系统，它可以克服LP带的缺陷。表1.2比较了LP和CD的重要特征（Bloom, 1985）。

在CD中，利用数字形式随着螺旋轨道记录信息，螺旋轨道是由连续的凹痕（pit）组成的（参见图1.12）（Carasso et al., 1982）。在CD上记录的每一位占有1 μm^2的面积，也就是每平方毫米有10^6位，所以CD上信息的密度非常高。

在录制期间，CD中音频信号处理的简化框图如图1.13所示。立体声通道的每一路模拟音频信号以44.1 kHz进行抽样和数字化，每一个抽样值用16位代码表示，表示90 dB的动态范围。这样，在每一个抽样间隔得到32位，左通道和右通道各16位。数字抽样值用两级Reed-Solomon编码方法编码，这样可使检测和校正的误差达到最小，或者在音频信号重放时隐藏的误差达到最小。为了便于听者控制以及显示信息，需要附加位。然后对得到的数据比特流进行调制，转换成更加适合唱盘存储的形式。EFM（8~14位调制）方法将数据流里的每一位转换为14位代码，得到信道比特流，

在进一步处理后用来控制激光波束，使数字信息记录到正在旋转的唱盘的光敏层上。利用照相的显影过程在主盘上产生凹痕图案。通过这个母盘，用户的激光唱盘随后可以生产出来。

表 1.2 LP 记录和激光唱盘（CD）的特征比较

特征	LP 记录	激光唱盘
频率响应	30 Hz 到 20 kHz（±3 dB）	20 Hz 到 20 kHz（+0.5 ~ −1 dB）
动态范围	70 dB（1 kHz 的速率）	>90 dB
信噪比	60 dB	>90 dB
谐波失真	1% ~ 2%	0.004%
立体声道之间的隔离	25 ~ 30 dB	>90 dB
抖晃	0.03%	检测不到
灰尘、刮伤和指印的影响	引起噪声	可校正或可隐藏
耐用性	Hf 响应随着播放变差	半持久的
唱针	500 ~ 600 小时	半持久的
播放时间	40 ~ 45 分钟（双面）	50 ~ 75 分钟（可扩展）



图 1.12 激光在唱盘上刻出的凹痕，每一个凹痕有 0.5 μm 宽、0.8~3.5 μm 长以及 0.11 μm深，轨道之间的距离为 1.6 μm（*Philips Technical Review*, **40**(6), 1982）

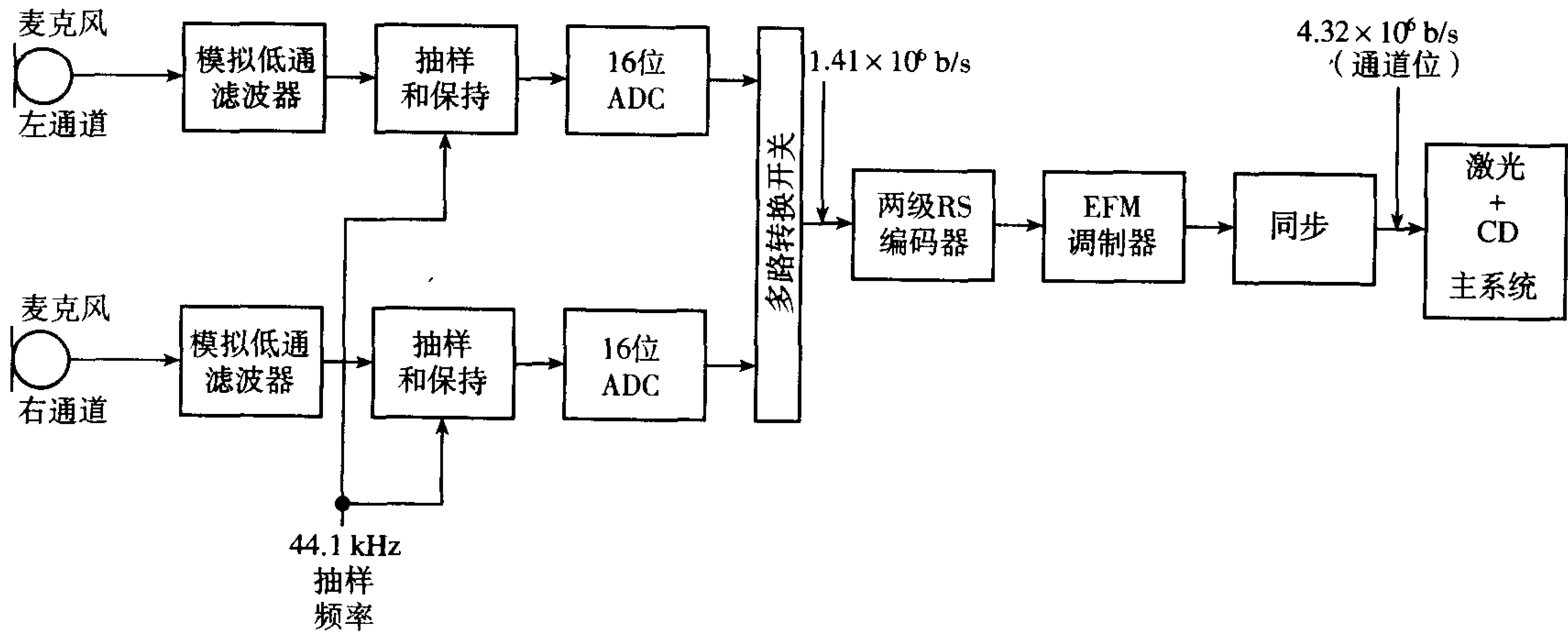


图 1.13 激光唱盘系统中音频信号的处理和记录的简化框图

在CD播放器的重放过程中，为了读出记录的信息，当唱盘以8转/秒和大约3.5转/秒之间的速度旋转时，唱盘上的轨道以1.2米/秒的恒定速度进行光学扫描（参见图1.14）。来自唱盘上的数字信息首先被解调，然后检测到数据中的任何错误，如果可能将加以校正。错误可能是由于制造上的缺陷、损坏、指印或者唱盘上的灰尘引起的。如果错误是不可校正的，那么通过用邻近的正确抽样值经内插来取代错误的值；或者如果错误不止一个抽样值，则将其设为零（静音）。

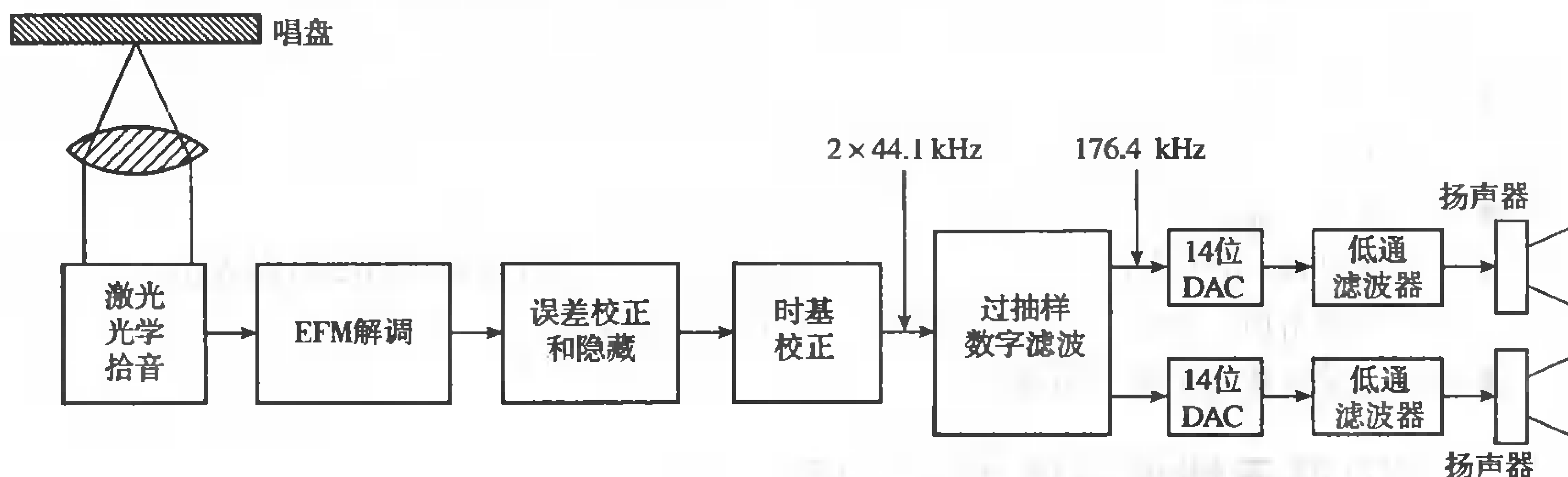


图1.14 在激光唱盘系统中音频信号的播出

在误差校正或者隐藏以后，得到的数据是一串16位的字，每一个字表示一个音频信号抽样值，这些抽样值可以直接应用到16位的DAC，然后进行模拟的低通滤波。然而，这要求模拟滤波器具有非常严格的技术规范，特别是20 kHz以上的频率电平相对于最大音频信号应该至少减少50 dB，滤波器在音频带内应该具有线性相位特性，以避免损害声音波形。为了避免这一点，让数字信号通过工作在音频抽样频率44.1 kHz的四倍的数字滤波器，从而得到进一步的处理。增加抽样频率的效果就是使DAC的输出更加光滑，减少模拟滤波器的要求，这也有利于用14位的DAC达到16位信噪比的性能。数字滤波器的应用允许达到线性相位响应，减少交叉调制的机会，从而得到一个具有随时钟频率变化特性的滤波器，使它对唱盘的旋转速度不敏感。图1.15给出了译码电路的印刷电路板，这是第一代Philips CD播放器，关键的IC已经清楚地标记出来。

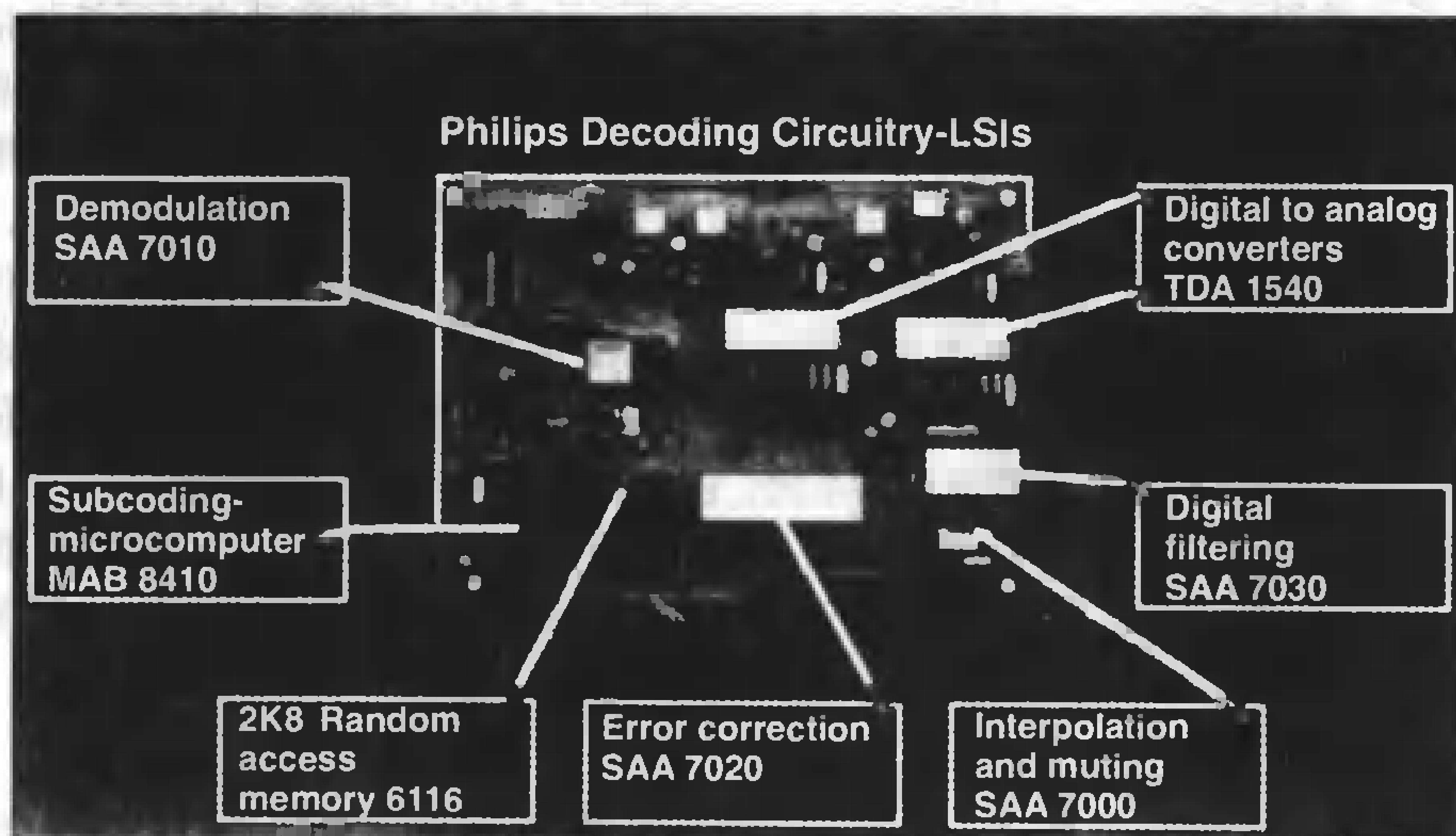


图1.15 Philips CD播放器译码电路的印刷电路板（*Philips Technical Review*, 40(6), 1982）

除了CD，DSP在其他数字音频工程中也起到了非常重要的作用，它广泛应用于消费产品或专业音频工作中，如录音棚、TV节目通过广播当局的传送和发行、电影以及音乐工业。DSP在数字音频工作中的特定应用包括（有些已经提到）：

- 将先进的DSP技术应用到编码、检测和校正或者隐藏由信号的失落引起的错误，在重播时消除抖晃，确保由记录媒体（电磁或光）带来的限制在记录和播放时不再影响可达到的质量。这样，对于具有类似错误率的不同牌子的带子，记录器的输出听起来都是相同的。
- 增强收听的环境并丰富声音。例如，简单的数字滤波器结构用来创建回声、自然的混响和合唱效果。
- 合成模仿乐器的声音或者其他乐器不能产生的声音。
- 在商业电视、卡通片和电影中，为了增强现实感或者增加情景的可信度而创造和运用声音效果，如射击声、脚步声、鼓掌欢呼声、汽车声、打孔声。
- 档案记录或法庭记录的增强。

1.7 DSP在无线电通信中的应用

1.7.1 数字式蜂窝移动电话

1.7.1.1 引言

移动通信是当今世界增长最快的工业之一，移动电话已成为信息社会保持接触的必不可少的工具。据估计，在今后几年内，世界上移动电话用户的数量将超过固定电话用户的数量，这在像芬兰等一些国家已经达到了。DSP是使移动电话革命成为可能的关键技术之一，DSP广泛地用于在基站和移动电话自身中处理信号和数据（如语音编码、多径均衡、信号强度测量、话音通信、误差控制编码、调制和解调）。适合于无线通信的DSP芯片现在可现货供应，它使移动通信工业为大众市场提供了可购买的高质量的产品。

现代移动电话系统使用数字蜂窝无线电话台的概念，但是，为蜂窝无线电电话打下基础的第一代移动电话系统使用了模拟技术来处理和传送话音信号。最成功的模拟移动电话系统包括在北美使用的高级移动电话系统（Advanced Mobile Phone System，AMPS），由丹麦、芬兰、挪威和瑞典联合开发的北欧移动电话系统（Nordic Mobile Telephone System，NMTS），以及在英国使用的全接入通信系统（Total Access Communication System，TACS）。早期的系统彼此是不兼容的，大都是由一些特定国家设计的，因此没有足够的能力来满足迅速增长的移动通信系统的需求。现代数字蜂窝电话网络提供了更大的容量、较大的覆盖范围以及高质量和安全可靠的通信方式。我们选用移动通信全球系统（Global System for Mobile Communication，GSM）来说明移动通信的有关问题。

GSM是第一代全数字蜂窝无线电电话系统，它现在被认为是移动数字通信的事实上的世界标准。GSM是1992年投放市场的，到1998年末，在世界100多国家已有超过1.3亿的用户。预计到2005年，移动电话用户将超过10亿，而这一数字的大部分是GSM用户。在欧洲使用的GSM 900的某些特征总结在表1.3中。

1.7.1.2 蜂窝电话网络的结构

移动蜂窝无线电网络是双向的电话网络，它允许移动电话通过无线电链路发射和接收信息（语音、数据和信息）。在蜂窝无线电系统中，覆盖的区域被划分成称为小区（cell，无线电小区）的单元，每一个小区由无线电基站提供服务，可用的频带由无线电频率或信道分开，每个小区分配一组无线电频率。为了有效地利用频带，无线电频率在覆盖的范围内复用，如图1.16所示（Macario, 1991,

1996)。为了使来自其他基站的、使用相同无线电频率的共同信道(co-channel)干扰最小，具有相同无线电频率组的单元应尽可能地分开。

表1.3 GSM 900的基本特征

蜂窝移动系统的参数	GSM 900的特性参数
频带	
移动发射/基站接收	890~915 MHz
移动接收/基站发射	935~960 MHz
双工频率分隔	45 MHz
信道频率间隔	200 kHz
信道数	124
语音速率	13 kb/s(半速率5.6 kb/s)
总的比特率	270 b/s
无线电传输方法	窄带CDMA
典型的小区尺寸	300 m ~ 35 km

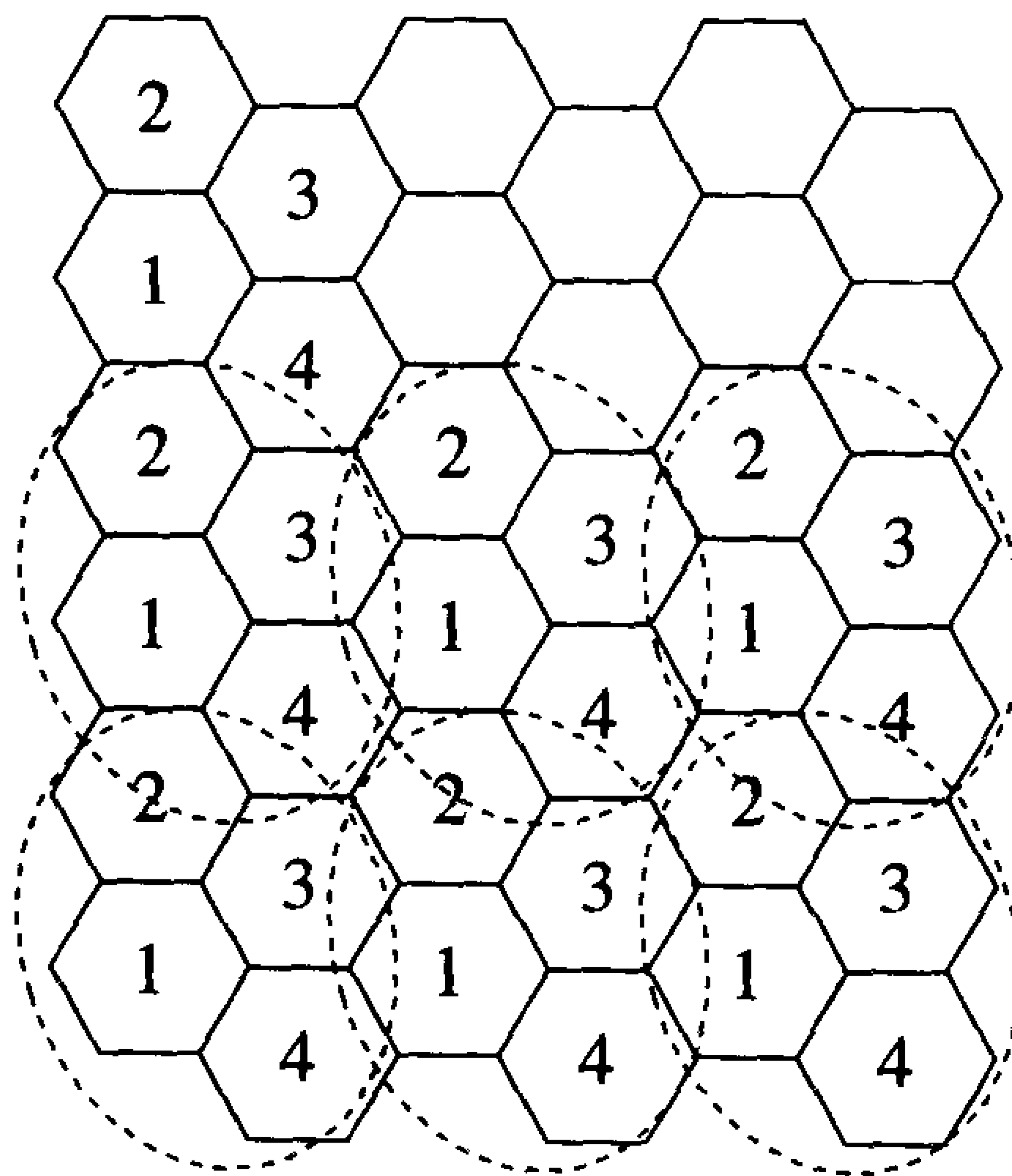


图1.16 蜂窝概念用频率复用的结果来说明小区重复模式

在实际应用中，小区尺寸根据话务量密度来改变。在城市和人口稠密地区，小区尽可能地小以满足高话务量水平(常常采用小于300米的微型小区)。在低话务量水平的农村，可能采用尺寸为35 000米的小区尺寸。这种大的小区要求大的传输功率，并且在覆盖的区域内可能包含有空穴。

通信是通过无线电基站在移动电话和网络之间发生的，蜂窝网络将无线电基站与覆盖范围内的单个无线电电话系统捆绑在一起。每个基站都具有一定数量的语音信道，并且连接到移动交换中心(MSC)，请参看图1.17(Macario, 1991; Horrocks and Scarr, 1993)。有一些MSC自身连接到PSTN，在蜂窝网络与PSTN之间提供网关。无线电基站由许多基站控制器(用于管理无线电信道)组成，每一个控制器具有一组基本收发器，MSC维持对移动电话位置的记录，并且对他们的移动进行管理。

移动蜂窝无线电网络的两个关键特征是移动电话位置确定和电话从一个基站移到另一个基站时无线电链路的切换能力。网络连续地记录每个登记电话的位置，以便在覆盖范围内呼入和呼出时能找到路线。这是十分重要的，因为移动电话用户并不在固定的位置，当移动电话开机时，他在网络上登记，这样可以使网络更新用户的位置。每一个移动电话在网络上至少有两个位置寄存器，一个归属位置寄存器(HLR)和一个拜访位置寄存器(VLR)。HLR是在用户的本地网络上并保留用户

的有关管理信息，如电话的访问业务。VSR含有移动电话在本地网之外使用的有关信息。当移动电话在网络登记时，要求它通过最近的基站向网络传递有关电话的惟一代码信息，然后从移动HLR得到必要的信息来鉴别和准许对网络的移动电话接入。

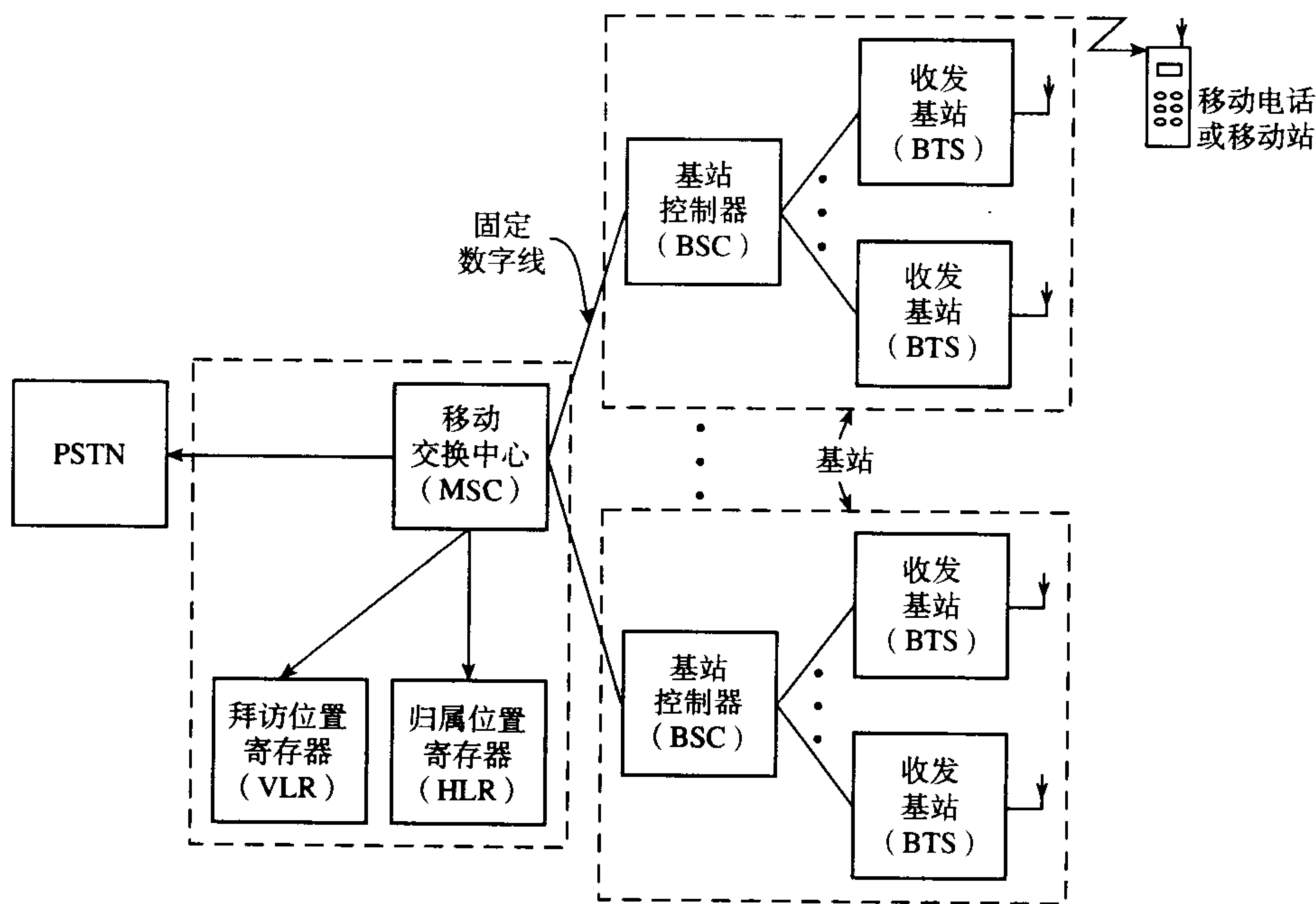


图 1.17 移动蜂窝电话系统的简化框图

正如前面所指出的，移动电话和网络之间的无线电链路是通过基站提供的，所有的无线电基站有规则地发射代码控制信息。在一次呼叫中没有接入的移动电话锁定到最近基站的控制信道上，当移动电话在覆盖区域内漫游时，它将被锁定到一个新的基站，它的位置将被更新。在一次呼叫中，将连续地监控接入的移动电话的信号强度，当信号的强度落到某个门限之下时，网络将无线电连接切换到另一个能够提供较好接收的基站（如果存在）。当信号强度很弱时，无线电链路从一个基站切换到另一个基站的能力允许移动用户在覆盖区域内自由地漫游，并且仍然能够打出和接听。切换过程要花几秒钟，尽管在通话时可能只注意到200~300 ms的中断。正如大多数移动电话用户知道的那样，在某些遥远的区域，刚好在附近不存在其他能够提供较好接收的基站。当信号强度太弱时，话音信道被切断，用户就不能打出或接听。用户可以根据显示形式或者移动电话显示屏上的其他内容（如信号条棒的数目）来推测信号强度。

1.7.1.3 信号处理方面

现代蜂窝无线电电话系统（如GSM）使用了数字技术，所以DSP是处理和传送信息的自然选择。在移动无线电电话系统中，DSP用于语音编码、多路径均衡、信号强度和质量测量，以及话音传送、误差控制编码、调制和解调（Macario, 1991）。

在GSM中，语音CODEC（多媒体数字信号编解码器）是基于规则脉冲激励，即线性预测编码（RPE-LPC）。不像PSTN语音是用64 kb/s或32 kb/s（自适应ADPCM）的速率编码，在移动无线电电话中，为了有效利用无线电频谱，语音是用相对较低的速率即13 kb/s进行编码的。GSM的语音编码算法已经用最流行的DSP处理器实现（如摩托罗拉的DSP56000，德州仪器的TMS320C50），多媒体数字信号编解码器有13 kb/s的基本数据率，并且提供了等价的13位线性ADC和DAC。

在移动通信中，由于移动电话操作的不利环境，常常会遇到多路径传播问题。在蜂窝无线电电话使用的频率点，发射的信号常常从高层建筑发射，反射的信号被延迟，并且比直接的信号后到达接收机，传播了更长的距离。这导致在接收机中组合信号的幅度和相位起伏，具体情况则取决于多径的特性和移动电话的运动。多径传播的影响可以在接收机利用数字均衡来减少。一个26位长的已知序列以规则间隔发射，在接收机末端，均衡器使用训练序列调整数字滤波器的系数来估计无线电路径的特性，因而可以消除接收数据多路径的影响。传递函数的知识使得接收机能够确定最可能发射的位序列，因而可以解调信号，GSM均衡算法已经在多种DSP处理器上实现。

除了语音编码和多径均衡，DSP也可以在数字调制中找到应用，许多技术用来在GSM系统中调制载波，包括多抽样率DSP技术。DSP也提供了测量接收信号强度的手段来帮助切换和调整由基站提供的输出功率电平。在GSM网络中，移动电话监控来自周围基站的信号。在蜂窝无线电系统中，来自使用相同频率的小区的同信道干扰总是一个值得关注的问题，它是发射功率的函数。在GSM中，发射的功率可以自动控制，以便降低同信道干扰。增加通话时间和电池待机时间，对发射功率进行调整的判定是基于接收信号的电平和质量。DSP技术用来分析接收信号，以便能够评估这些参数。

移动无线通信系统会出现许多误差，例如由随机干扰和衰落引起的误差。在GSM中采用常规的编码技术来减少这些误差的影响。

1.7.2 数字电视接收的机顶盒

近年来，大规模的数字电视播送已经开始进行，这将给消费者带来很多好处，如交互性、更多的选择、好的图像和高质量的声音。交互性使用户能够玩游戏、上网、购物、即时重放等，TV已经建成为信息社会的非常基本的组成部分。

在数字TV中，数字信息（视频、音频和文本）可以通过卫星（利用卫星和现有的卫星碟形天线）、电缆（有线电视）和陆地设备（利用现有的TV发射机和天线）发送到家里的电视机中。我们家中大多数的现有电视机都能够接收到模拟传输信号，所以，为了接收数字TV需要使用数字解码器（机顶盒），请参见图1.18。机顶盒将数字信息转换成适合模拟电视机接收的形式，最新的电视机都内置有解码器。

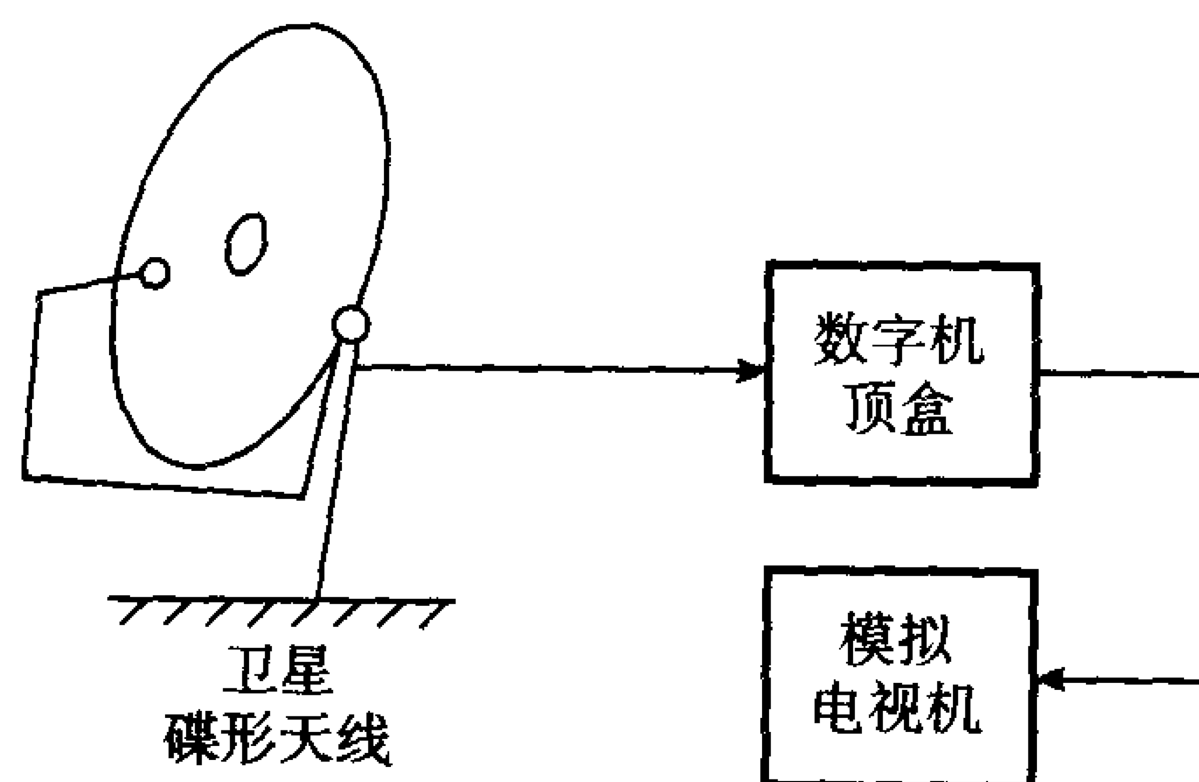


图1.18 数字电视接收机顶盒的示意图

在数字TV中，DSP在处理、编码/译码、视频和音频信号的调制/解调、从捕捉到信号直至在电视机上看到图像等过程中都起到了关键的作用（Benoit, 1997）。没有DSP，我们现在认为是理所当然的清晰图像和高质量声音都将是不可能的。例如，DSP是MPEG编码算法的核心，MPEG编码算法用于在传输前压缩视频和音频信息（充分地利用带宽）。在机顶盒中，MPEG解码器用来恢复信息。MPEG算法的一个关键部分是离散余弦变换，我们将在第3章中进行详细的介绍。

1.7.3　自适应回声对消

在通信系统中，当信号遇到阻抗失配时就会出现回声。在图1.19中给出了一个简化的远距离电话电路，在交换机处的混合电路将订户的两线电路转换成四线电路，对每个方向的传输提供分开的路径。出于经济上的原因，允许多路转换或者许多呼叫同时传输。

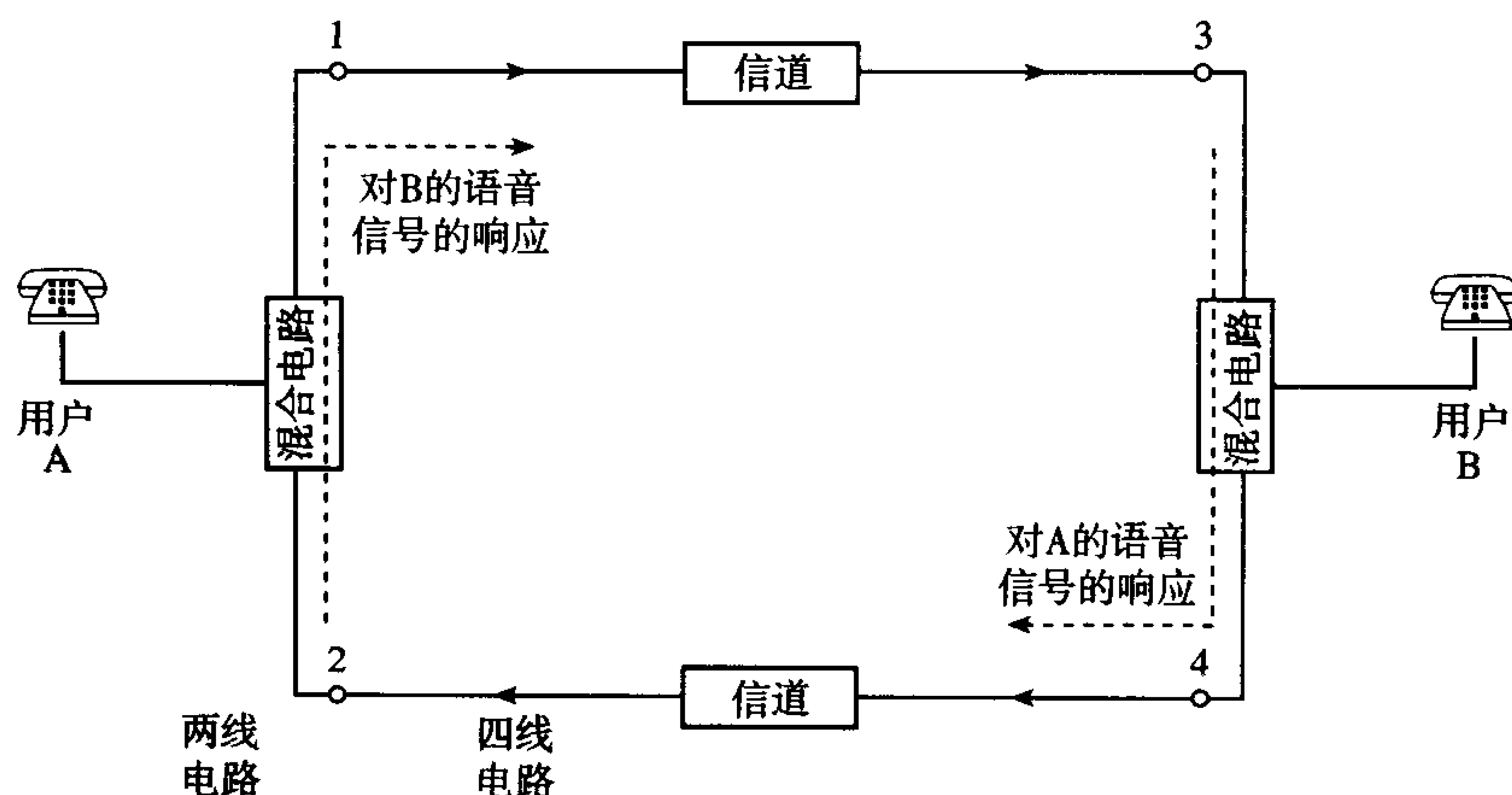


图1.19　简化的远距离电话电路

理想的情况是来自用户A的语音信号沿着上面的传输路径到达右边的混合电路，从那里再到达用户B，而来自用户B的语音信号沿着下面的传输路径到达用户A。在每一端的混合电路应该确保从远距离用户发来的语音信号耦合到它的两线端口，并且没有任何信号耦合到输出端口。然而，由于阻抗失配，混合网络允许某些输入信号泄漏到输出路径并且作为一种回声返回到说话者。打长距离电话时（例如利用同步卫星），回声可能延迟540 ms，这种回声是惹恼用户的一种干扰，这种干扰将随着距离而增加。为了克服这个问题，可以在网络中安装一对回声对消器，如图1.20所示（Duttweiller, 1978）。

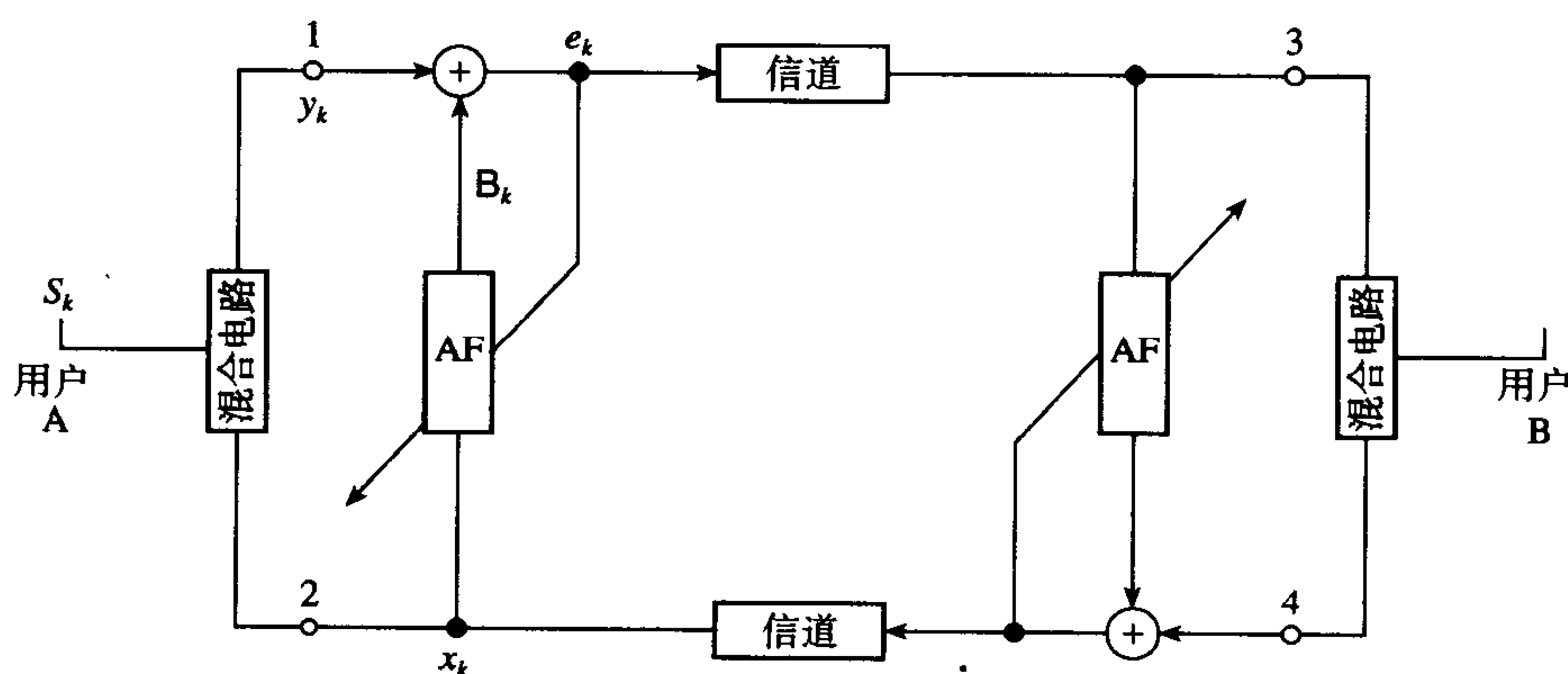


图1.20　在长距离语音通话中的回声

在通信系统的每一端（参见图1.20），输入信号x_k加到两个混合电路和自适应滤波器（AF）中，通过估计回声并且与返回的信号y_k相减就可以达到回声对消的目的，回声的估计由下式给出：

$$\hat{y}_k = \sum_{i=0}^{N-1} w_{k+1}(i) x_{k-i}$$

其中 x_k 是来自远端说话者的输入信号，$w_k(i)$（$i=0,1,...,N-1$）是回声路径的冲激响应在离散时间 k 的估计。

1.8 DSP在生物医学中的应用

生物医学是常规DSP应用和开发新的和稳健DSP算法的一个重要且广阔的领域。医学数据常常不是很规则，DSP实践者必须拿出处理数据的新方法，这是一种挑战。在大多数情况下，医学数据是在音频范围。因此，我们发现解决生物医学方面的问题的DSP技术在其他领域也可以找到，如声频、无线电通信和控制等，反过来也一样。

在生物医学方面的DSP应用都包括信号增强或者临床感兴趣的特征信息的提取。信号增强的需要是源于伪像（artefact）问题或者信号的污染，这样的问题在生物医学中是十分普遍的。伪像是由外部（例如，主要由电源引起的，以及其他医疗设备）和内部的（头和身体的移动、肌肉和心脏的活动以及眼的移动）因素引起的。伪像减少了生物医学信号临床的可用性，给人工分析和自动分析带来很大的困难，在某些情况下甚至是不可能的，这是因为伪像和临床感兴趣信号的相似性（Ifeachor, et al., 1990）。

信号增强任务常常由两个相似的问题所刻画，一个问题是信号电平与干扰噪声相比要低，另一个问题是信号与噪声谱的重叠。因此，通常予以关注的是要求通过信号增强以使临床感兴趣的信息的失真最小（Qutram et al., 1995; Wu et al., 1997）。下两节将描述在生物医学方面的两个新的DSP应用，在这两个应用中包含有信号增强和特征提取的问题。在这里，我们感兴趣的基本生物信号是生理信号，特别是心电图信号——心脏的电行为特征，以及脑电信号——大脑的电行为特征。

1.8.1 胎儿ECG监控

胎儿心电图（ECG）反映了从人体表面测到的胎儿心脏的电行为特征（Qutram et al., 1995）。胎儿的心率（FHR）根据ECG（参见图1.21）的R到R的时间间隔推导出来。FHR与子宫收缩（子宫活动）一起连续显示（称为胎心与分娩力描述图（CTG））的可视分析通常在分娩时用来评估胎儿的状况。在分娩期间CTG解释的困难可能导致不必要的医疗上的干预（如剖腹产、产钳分娩）、胎儿受伤，或者当需要干预时干预失败（Keith et al., 1995）。

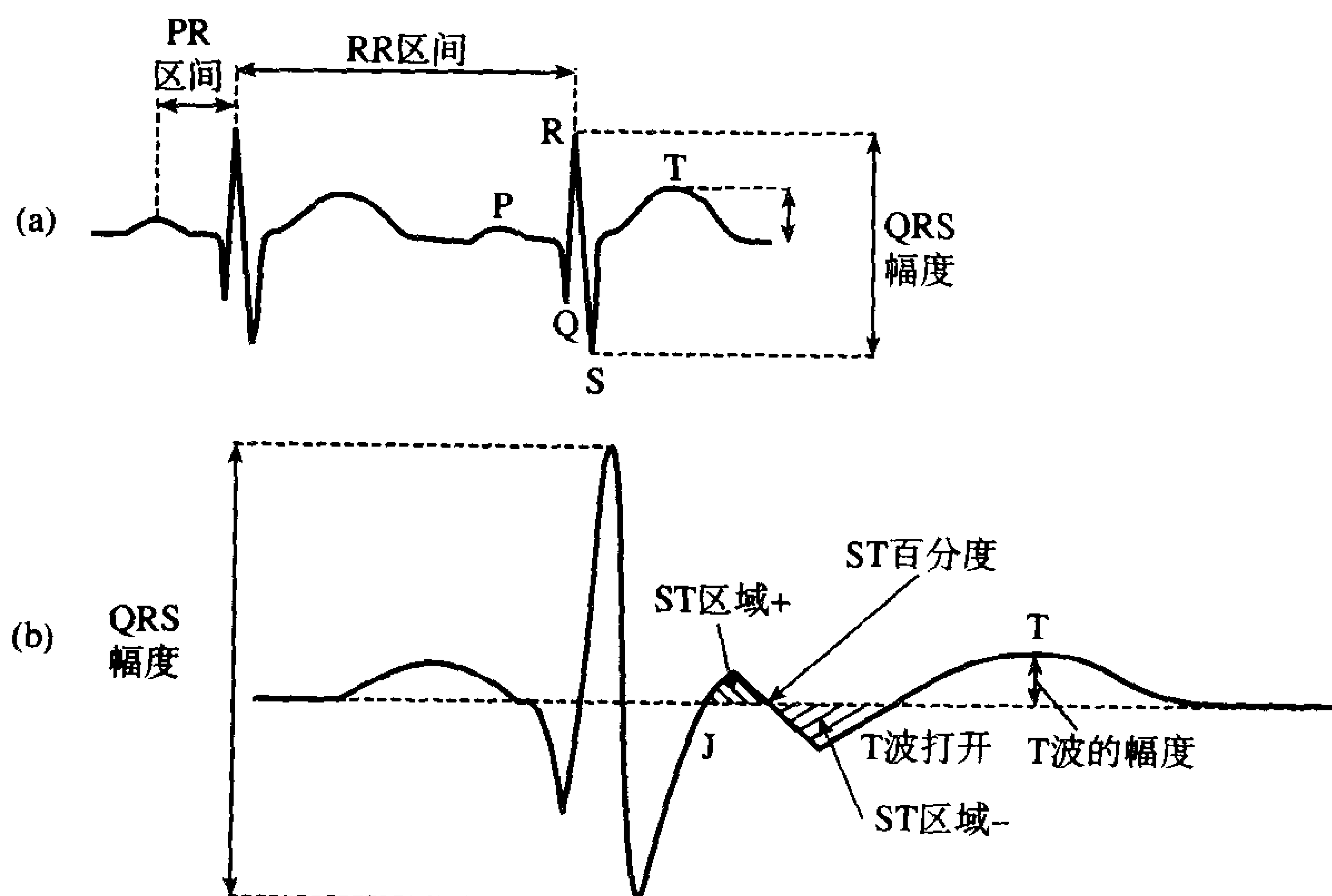


图1.21 胎儿ECG显示的临床有用的关键特征

将胎儿ECG和CTG组合在一起分析的正确使用将大大减少不必要的医疗干预，对胎儿的分娩不会产生不利的影响（Westgate et al., 1993）。使用这种组合分析的一种商业化的胎儿ECG监护仪、ST分析仪（STAN, Neoventa AB, Sweden）已经开发出来。

图1.22给出了一种基于ECG的简化的胎儿监护系统。为了得到好的信噪比，ECG是从头皮电极得到的，带宽限定为0.05 ~ 100 Hz，按每秒500个抽样点以12位的精度进行量化。为了减少噪声和特征提取，对胎儿ECG数据进行处理，然后定量地分析和显示与CTG相关联的波形的变化。

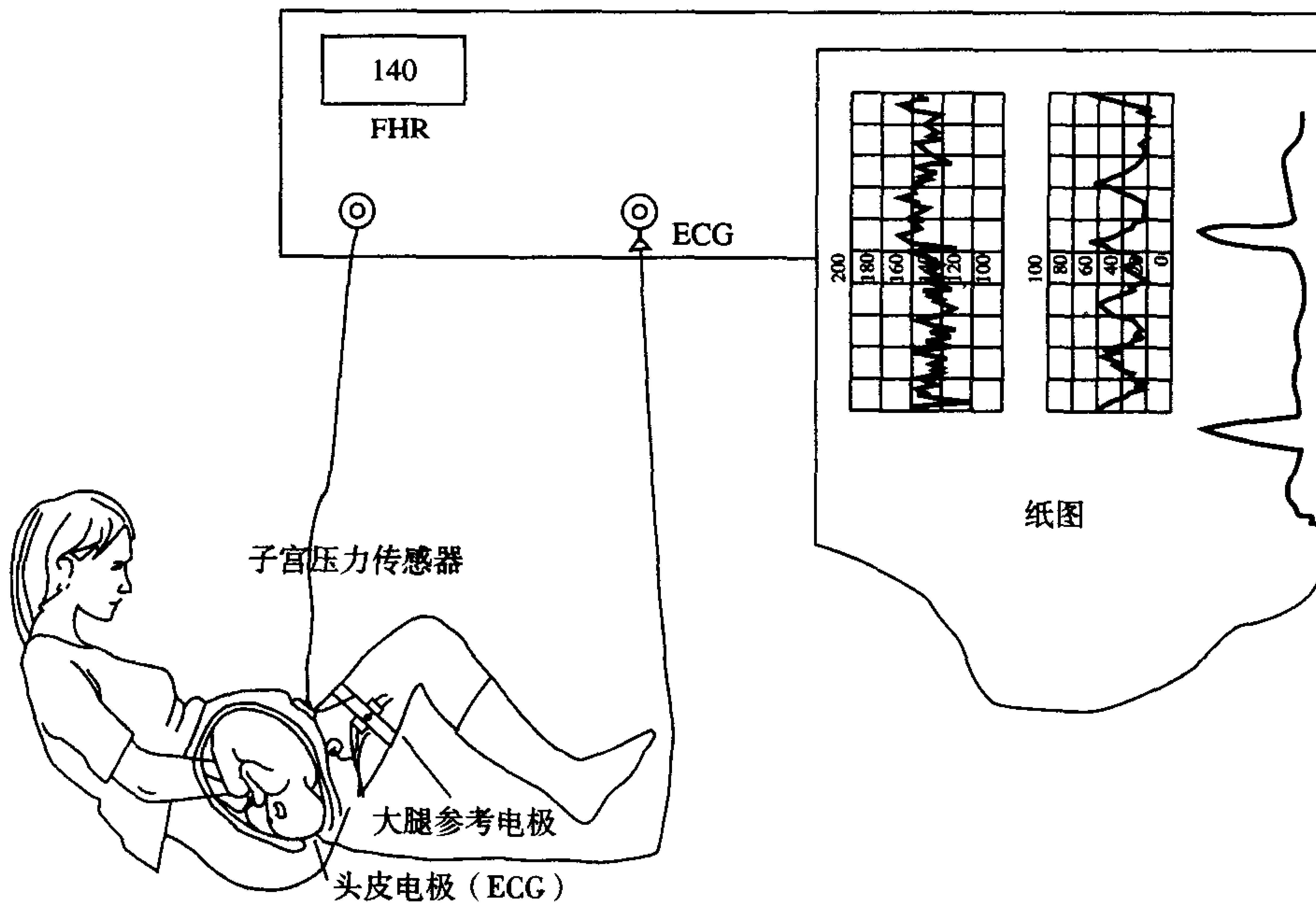


图1.22 分娩过程中胎儿ECG监护

胎儿ECG的一个很重要的特征是ST段的形状，与压力或痛苦有关的在形状上重要的模式变化包括持续上升的T波幅度、负T波和降低的ST段。ST波形的变化可以用T波的幅度与QRS的幅度之比来定量地表示，称为T/QRS比，请参见图1.21。其他有用的一些特征包括ST区域、R到R间隔的变化、P波的持续时间以及QRS的宽度。

胎儿的头皮ECG是很容易受到低频噪声及其他伪像的影响，其他伪像可能包含波形的虚假变化，如噪声、基线位移、肌肉伪像和随机噪声。伪像妨碍了特征的提取，可能会导致不精确的ECG特征和波形分析的结果。

有许多信号处理方法（包括曲线拟合和多抽样率滤波）可以用来减少噪声，并且从ECG中提取关键特征，图1.23给出了胎儿ECG信号处理的框图。第一个主要的任务就是精确地检测R波，从原始的ECG中必须移去基线位移、肌肉噪声和电力线频率，以得到一个适合于可靠分析的波形。图1.24给出了一个基线位移估计和消除的例子。

很显然，如果没有DSP，基于胎儿ECG的监护是不可能的。然而，尽管前面提到的妨碍胎儿ECG分析的信号处理问题已经解决，但是基于ECG的监护仍然限制在少数几个中枢。目前，与CTG有关的ECG变化的分析和解释是通过可视化的检查来进行的，临床医生要用好胎儿ECG需要进行训练。但是，如果没有专家监督，既使训练也仍然有可能漏掉重要的ECG图形。目前正在开发连续地分析和解释ECG和CTG变化的床边智能监护仪（Keith et al., 1995, Ifeachor and Outram, 1995）。

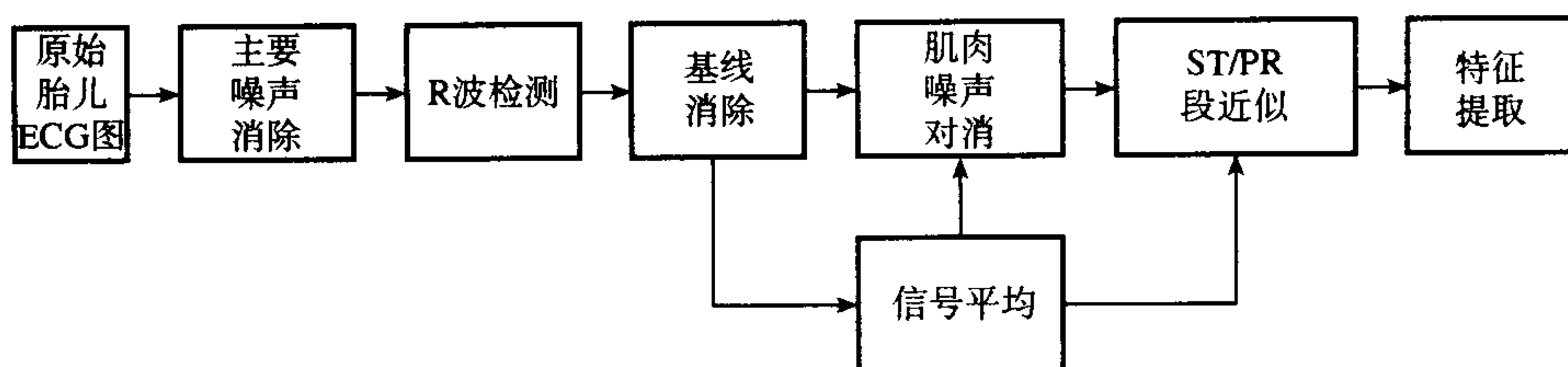


图1.23 胎儿ECG信号处理框图

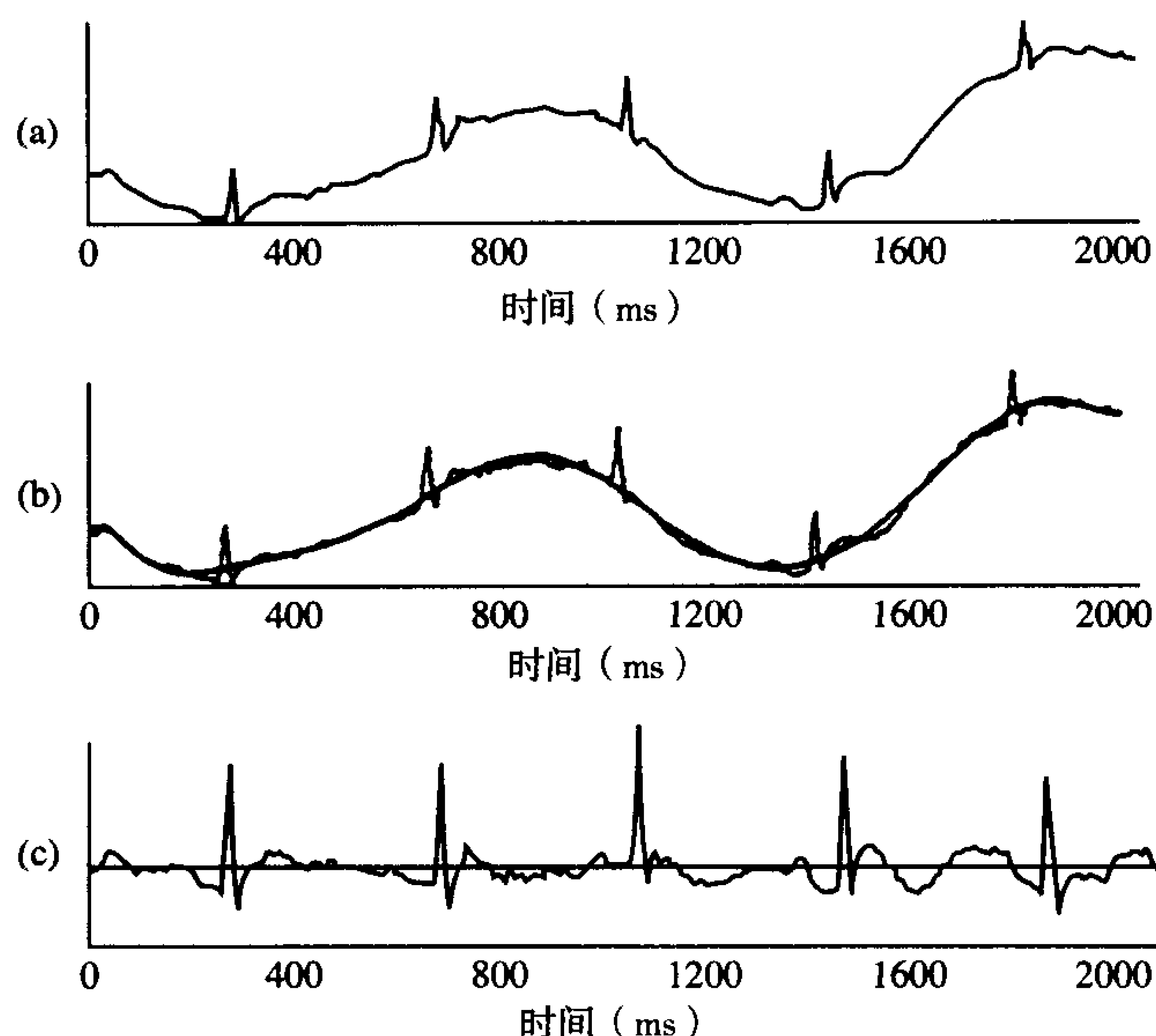


图1.24 (a)近似；(b)从原始胎儿ECG中移去基线位移

1.8.2 基于DSP的闭环控制麻醉术

DSP在所有主要医院的加护病房中有着广泛的应用，许多先进的技术正在连续不断地开发以应用于麻醉术。在手术的过程中，病人通常要进行麻醉，例如静脉注射麻醉药，这样在手术过程中病人不会感到痛苦，从而创造一种外科医生操作手术的条件。麻醉师的目标就是要给出适当的药量来使麻醉药尽可能快地进入到一定的深度，并且维持这种水平直到有必要改变时为止。给病人注射过量的麻醉药可能导致并发症和其他副作用，而药物不当也可能导致在外科手术进行中出现某种意识，这可能造成长期的心理负担（Huang et al., 1999）。在大多数情况下，病人意识的深度是由有经验的麻醉师通过临床信号的观测来测定的，然后，麻醉师通过使麻醉药的剂量进行适当的变化来控制麻醉。采用闭环控制技术的自动药物传递对忙碌的麻醉师带来的潜在益处，则是可以提供更好的监护，并减少成本。该技术的应用减少了超剂量的可能性，使麻醉师能够迅速识别和响应可能没有注意到的摄动，或者认为其太小以至于不值得人为改变的小的变化。

然而，为了确定维持一定的麻醉深度所必需的药剂传递，要求可靠的自动闭环控制技术来监控麻醉深度。目前的闭环控制麻醉系统使用生物信号来测量麻醉的深度，并且把它作为一种反馈信号来确定药剂传递所需要的调整，特别是采用了许多信号处理方法来处理EEG信号，提取像听觉的诱发反应（AER）、双谱指数等特征，并根据这些特征参数来估计麻醉的深度（Huang et al., 1999）。EEG是大脑的电活动，它是通过放在头皮上的电极来测量的。AER是大脑对外部声音激励的电响应，AER信号对于评定从有意识到无意识的转换是有价值的。但要得到它却很困难，因

为它隐藏在EEG信号中，要比EEG信号小好几倍，为了提取它，常常采用连续的声音激励对响应信号取平均的方法。因此，为了从EEG背景信号中提取AER信号，需要对AER信号进行处理，以便提取临床感兴趣的特征（如峰值、反应时间和形状）。双谱指数是通过EEG信号的高阶谱分析推导出来的（Nikias and Raghuveer, 1987），它提供了不同意识级别下在EEG信号中的频率分量的复杂变化以及内部关系的定量的度量，已经知道它与propofol（一种最常见的麻醉药）的血液浓度呈线性相关。

图1.25给出了基于EEG的闭环控制麻醉（CLAN）系统的简化框图（Dong et al., 1998, 1999），系统的关键部分是获取原始EEG信号的EEG分析仪，它通过一个双峰电极连接到病人。在分析仪中采用了许多信号处理方法来减少噪声、提取特征、分析特征的变化以及计算合适的EEG指数。这些信号处理方法包括小波变换、信号平均、双谱分析和神经网络等。

计算的EEG指数（如双谱指数）给出了对病人注入药物效果的度量，计算的EEG信号则作为反馈信号与目标EEG指数进行比较，以确定目标血液浓度$C_T(k)$的变化。市场上出现了计算双谱指数和其他指数的EEG分析仪（如A-1000，Aspect Medical System）。药物动力学/药效（PK/PD）模型确定了药剂注入到病人的速度$I(k)$，这个速度是目标血液浓度和病人特征参数的函数，病人特征参数包括年龄、体重、性别等。静脉内的药剂常常通过一个合适的麻醉泵在目标控制注入程序（如STANPUMP®, S.L. Shafer, Stanford University）的控制下注入到病人体内。图1.26给出了一个麻醉期间双谱变化的例子。

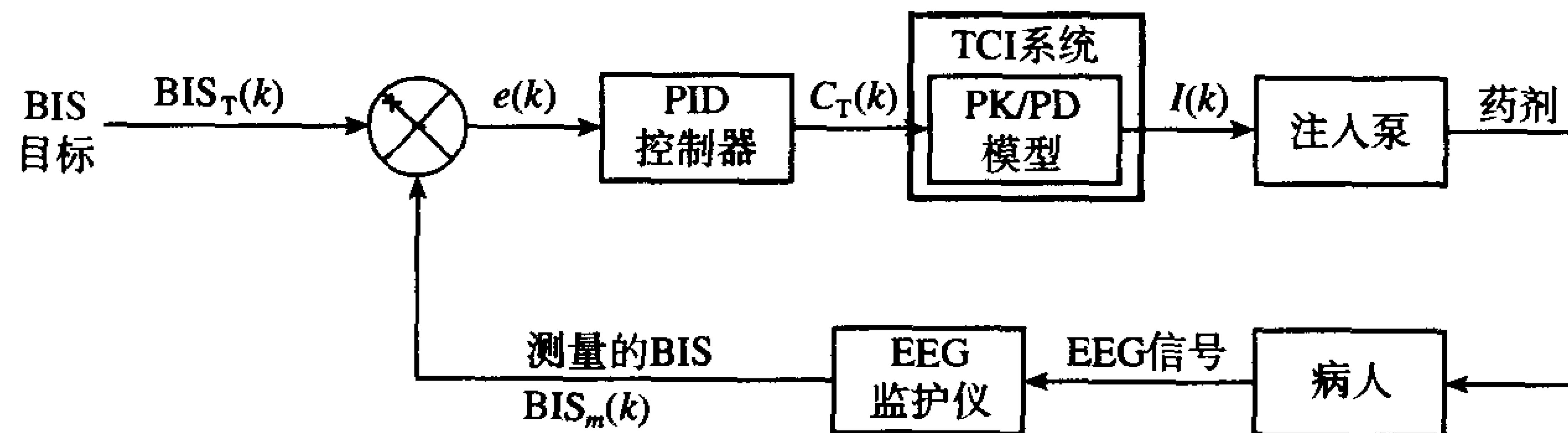


图1.25 基于DSP的闭环控制麻醉系统

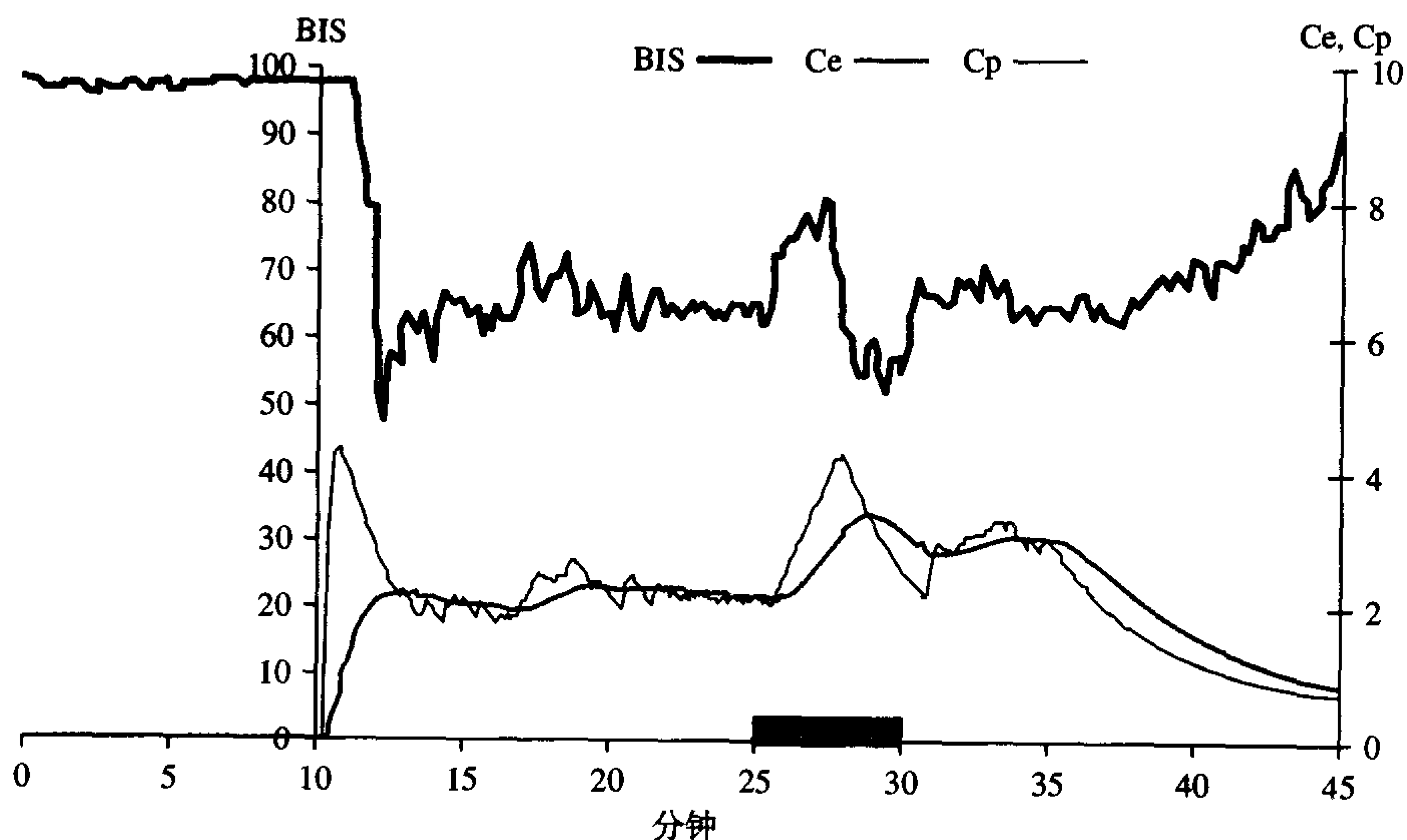


图1.26 志愿者在闭环控制麻醉期间双谱变化的一个例子

1.9 小结

在本章我们讨论了应用领域中的DSP方法，并且确定了关键的DSP运算。

从我们讨论的特定的应用例子中可以看出,DSP已经在消费和专业的电子领域中产生了重要的影响。

习题

1.1 论述DSP与模拟信号处理系统设计相比的两个主要优势和两个主要劣势。

1.2 利用框图描述激光唱盘播放器音频信号播放的过程,并叙述和证明在该应用领域采用DSP技术的四大优势。

1.3 GSM在今天代表什么？解释GSM在过去代表什么？

1.4 借助框图描述移动蜂窝无线电电话是如何工作的？

1.5 简单地描述DSP在数字蜂窝无线电电话中的作用，为什么在数字蜂窝无线电电话系统中DSP是处理语音信息的自然选择？

1.6 解释无线电小区的含义，大概地画出三个小区重复图。

参考文献

Advanced Mobile Phone Service. *Bell System Technical Journal*, **58**(1), January 1979.

Bellamy J.C. (1982) *Digital Telephony*. New York: Wiley.

Benoit H. (1997) *Digital Television: MPEG-1, MPEG-2 and Principles of the DVB System*. London: Arnold.

Bloom P.J. (1985) High-quality digital audio audio in the entertainment industry: an overview of achievements and challenges. *IEEE ASSP Magazine*, October, 2–25.

Carasso M.G., Peek J.B.H. and Sinjou J.P. (1982) The compact disc digital audio system. *Philips Technical Rev.*, **40**(6), 151–6.

Clark R.J., Ifeachor E.C., Rogers G.M. and Van Eetvelt P.W.J. (2000) Techniques for generating digital equaliser coefficients. *Journal of Audio Engineering Society*, **48**(4), 281–98.

Dong C., Kehoe J., Henry J., Ifeachor E.C., Reeve C.D. and Sneyd J.R. (1998) Closed loop computer controlled sedation with propofol. *British Journal of Anaesthesia*, **81**, 631P.

Dong C., Reeve C.D., Sneyd J.R. and Ifeachor E.C. (1999) Closed-loop control of intravenous drug infusion. *IEE Proc. Sci. Meas. Technol.* (submitted).

Duttweiller D.L. (1978) A twelve-channel digital echo canceler. *IEEE Trans. Communications*, **26**, 647–53.

ETSI/GSM Recommendations: ETSI, BP 152, F-06561 Valbonne CEDEX, France.

Frantz G.A. and Wiggins R.H. (1982) Design case history: Speak and Spell learns to talk. *IEEE Spectrum*, February, 45–9.

Horrocks R.J. and Scarr R.W.A. (1993) *Future Trends in Telecommunications*. New York: Wiley.

Huang J.W., Lu Y., Nayak A. and Roy R.J. (1999) Depth of anaesthesia estimation and control. *IEEE Trans. Biomed. Eng.*, **46**(1), 71–81.

Ifeachor E.C., Hellyar M.T., Mapps D.J. and Allen E.M. (1990) Knowledge-based enhancement of EEG signals. *Proceedings of IEEE Radar and Signal Processing*, **37**, 302–10.

Ifeachor E.C. and Outram N.J. (1995) A fuzzy expert system to assist in the management of labour. *Proc. International ICSC Symposium on Fuzzy Logic*, ICSC, C97–102, Zürich, Switzerland.

Keith R.D.F., Beckley S., Garibaldi J.M., Westgate J., Ifeachor E.C. and Greene K.R. (1995) A multicentre comparison study of 17 experts and an intelligent computer system for managing labour using the cardiotocogram. *Brit. J. Obstet. Gynaecol.*, **102**, 688–700.

Macario R.C.V. (ed.) (1991) *Personal and Mobile Radio Systems* (Chapters 4, 9, 13 and 14). Peter Peregrinus Ltd for the Institution of Electrical Engineers.

Macario R.C.V. (ed.) (1996) *Modern Personal Radio Systems* (Chapters 3, 8, 11 and 12). The Institution of Electrical Engineers, London.

Nikias C.L. and Raghuveer M.R. (1987) Bispectrum estimation: a digital signal processing framework. *Proc. IEEE*, **75**, July, 869–91.

Outram N.J., Ifeachor E.C., Van Eetvelt P.W.J. and Curnow J.S.H. (1995) Techniques for optimal enhancement and feature extraction of fetal electrocardiogram. *IEE Proc. Sci. Meas. Technol.*, **142**(6), November, 482–9.

Watkinson J. (1994) *The Art of Digital Audio*. Second edition. Oxford: Butterworth-Heinemann.
Westgate J., Harris M., Curnow J. and Greene K.R. (1993) Plymouth randomised trial of the cardiotocogram only versus ST waveform plus cardiotocogram for intrapartum monitoring in 2400 cases. *Am. J. Obstet. Gynecol.*, **169**, 1151–60.
Wu J., Ifeachor E.C., Allen E.M., Wimalarantna S.K. and Hudson N.R. (1997) Intelligent artefact identification in electroencephalography signal processing. *IEEE Proceedings Science, Measurement and Technology*, **144**(5), 193–201.

参考书目

Mitra S.K. (1998) *Digital Signal Processing – A Computer-Based Approach*. New York: McGraw-Hill.
Mulgrew B., Grant P. and Thompson J. (1999) *Digital Signal Processing – Concepts and Applications*. Basingstoke: Macmillan.
Oppenheim A.V. and Schafer R.W. (1975) *Digital Signal Processing*. Englewood Cliffs NJ: Prentice-Hall.
Oppenheim A.V. and Schafer R.W. (1989) *Discrete-Time Signal Processing*. Englewood Cliffs NJ: Prentice-Hall.
Orfanidis S.J. (1996) *Introduction to Signal Processing*. Englewood Cliffs NJ: Prentice-Hall.
Papamichalis P. (1987) *Practical Approaches to Speech Coding*. Englewood Cliffs NJ: Prentice-Hall.
Rabiner L.R. and Gold B. (1975) *Theory and Applications of Digital Signal Processing*. Englewood Cliffs NJ: Prentice-Hall.

第2章　实时DSP系统的模拟I/O接口

在许多实际的应用中，信号是模拟形式的，而DSP是对数据进行操作。因此，为了使DSP系统与实际应用接口，我们需要模拟输入/输出（I/O）接口来允许模拟和数字格式的转换。

在实时DSP中，模拟I/O接口是一种弱的连接，因为它引入了难以复归的误差，并且对速度的限制也有影响（参考后面的内容）。很好地理解模拟I/O接口的设计问题，对于成功地设计具有模拟输入或模拟输出的实时DSP系统是必备的条件，对于其他DSP应用领域（如多抽样率处理）也是必要的。本章覆盖了许多设计问题，特别是在结束本章的学习时，读者应该：

(1) 理解实时DSP的模拟I/O接口设计的基本理论（如低通和带通抽样定理，如何将它们应用到实际问题中，以及出现在模拟I/O接口中的误差的性质）；
(2) 能够刻画、分析和确定模拟I/O系统的基本参数（如抽样频率、混叠误差的级别）；
(3) 理解模拟I/O接口过抽样的基本原理（如过抽样和噪声整形，简单的过抽样转换器的设计和分析）。

我们从音频、无线电通信和生物医学的应用来说明其原理。

2.1　典型的实时DSP系统

图2.1描述了实时工作的典型DSP系统的框图。模拟输入滤波器用于在量化前带限模拟输入信号，以便减少混叠（参考后面的内容）。ADC将模拟信号转换成数字形式。对于宽带信号，或者当采用慢的ADC时，在ADC之前加入抽样和保持电路是必要的。经过处理器中的数字处理后，DAC将处理过的信号转换成模拟形式，输出滤波器将DAC的输出进行平滑，并消除不要的高频分量。

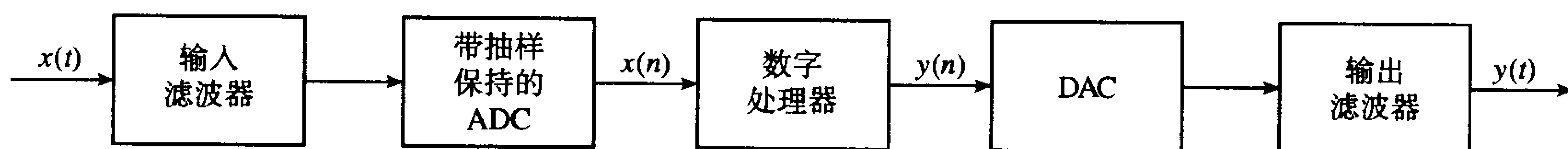


图2.1　简化的、一般性的实时数字信号处理系统的框图。在某些应用中，不需要输入滤波器和ADC或者DAC和输出滤波器

图2.1系统的核心是数字处理，它可能是一个基于像摩托罗拉MC68000的通用微处理器、德州仪器的TMS320C50数字信号处理器芯片或其他硬件块。数字信号处理器可以实现几种DSP算法，例如数字滤波，将输入$x(n)$变换成输出$y(n)$。

采用数字处理器的信号处理意味着在处理前输入信号必须是数字形式。在某些实时应用中，数据已经是数字形式，或者并不需要转换成模拟信号。例如，信号在处理后可以存在计算机内存中以便以后进行处理，或者可以在显示单元上以图形形式表示出来。在其他一些应用中，可能要求以数字形式产生信号。这样的例子有语音合成、数字频率合成和伪随机二元序列产生器。本书的许多讨论都假定信号是数字形式的或者已经经过下一节所描述的数字化。

2.2 模数转换过程

正如前面提到的，在任何DSP算法执行前，信号必须是数字形式。自然界的大多数信号是模拟形式，因此，模数转换过程是必要的，它包括下列步骤：

- （带限）信号首先被抽样，将模拟信号转换成时间离散幅度连续的信号；
- 每个信号抽样的幅度被量化成2^B个电平之一，其中B是ADC中用来表示一个抽样值的位数；
- 将离散幅度电平表示成或编码成B位长度的不同的二进制字。

图2.2描述了抽样的过程，在图中可以看出三个不同类型的信号：

- **模拟输入信号** 这个信号在时间和幅度上都是连续的；
- **抽样后信号** 这个信号在幅度上是连续的，但只定义在一些离散的时间点上。因此，信号除了$t = nT$（抽样时刻）外为零；
- **数字信号，$x(n)$（$n = 0, 1, \ldots$）** 这个信号只在离散时间点上存在，每个时间点上只有2^B个值中的一个（离散时间离散值信号），这是本书我们所关心的信号。

注意，离散时间（即已抽样的）信号和数字信号的每一个都可以表示为一个序列$x(nT)$，或者简写为$x(n)$（$n = 0, 1, 2, \ldots$）。现在，让我们仔细考察一下数字化一个信号的每一步。

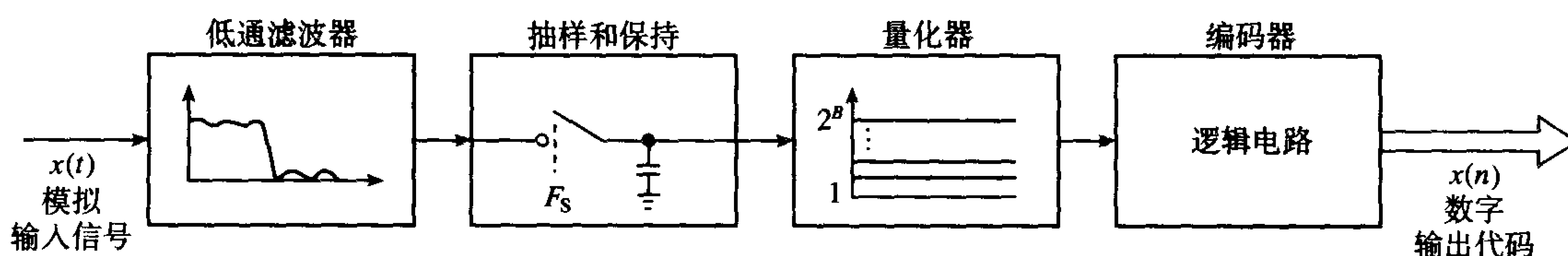


图2.2 模数转换过程的图解表示

2.3 抽样－低通和带通信号

抽样是在离散的时间间隔对连续时间信号（例如模拟信号）的采集，它是实时信号处理中的基本概念。图2.3给出了一个抽样后的模拟信号例子。注意抽样之后，在这种理想的情况下，模拟信号由一些离散时间的值来代表，这些抽样的值等于原始的模拟信号在离散时间点的值。

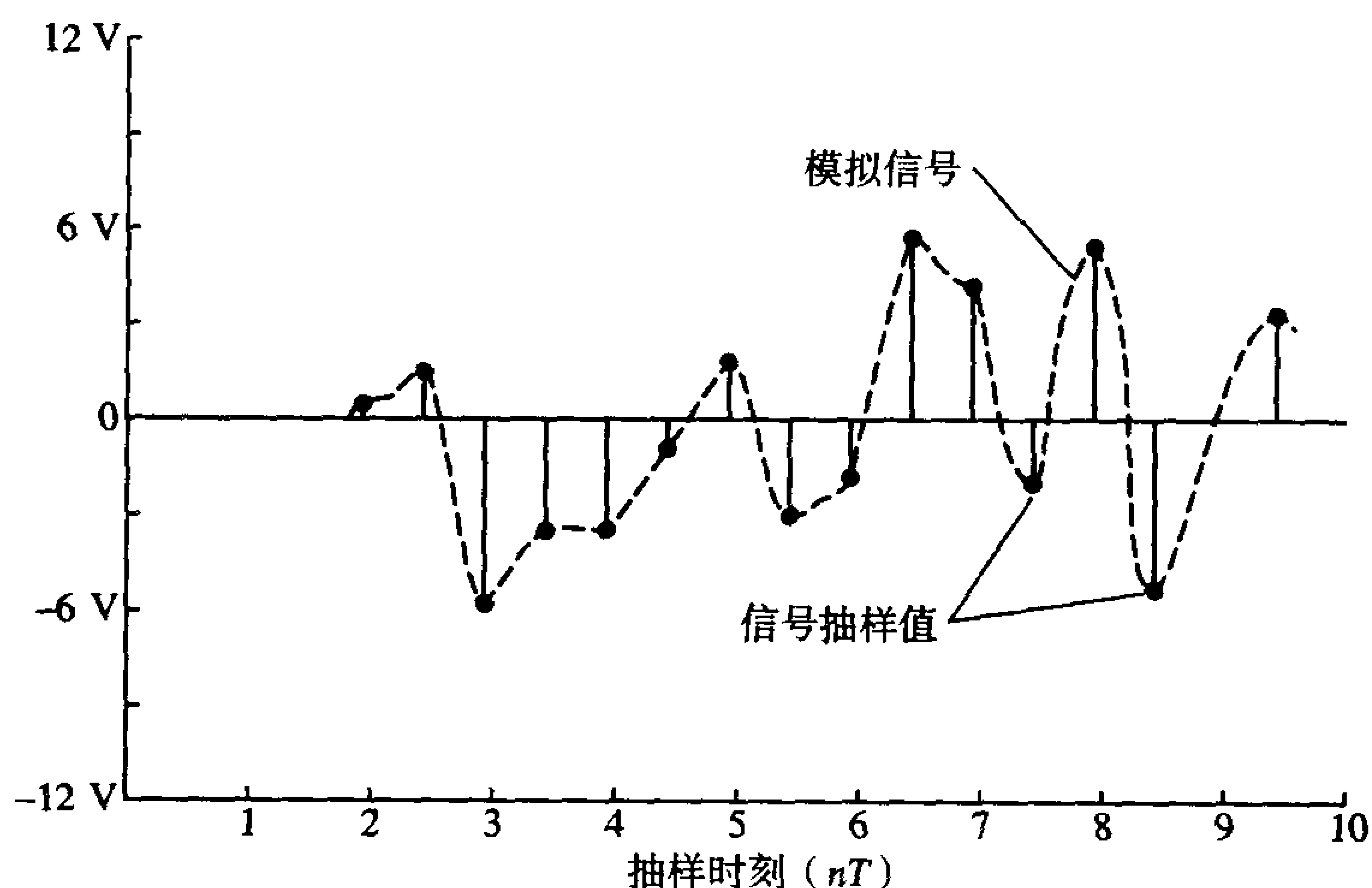


图2.3 信号抽样（理想抽样）的一个例子。信号抽样值等于原始的模拟信号在抽样时刻的值

在本章，我们将给出抽样定理的一个直观的描述，抽样定理规定对模拟信号应该以多大的速率抽样，以保证能够捕捉到包含在信号中的相关信息或者经过抽样后能够保留相关的信息。在实际应用中，我们常常遇到两种形式的抽样，低通信号的抽样和带通信号的抽样。后面我们将会看到，对于带通信号的抽样，可以将其看作为更为一般的低通抽样的特殊情况。

2.3.1　抽样低通信号

2.3.1.1　抽样定理

如果信号的最高频率分量是f_{max}，为了使抽样值能够完整地描述信号，那么至少应该以$2f_{max}$的速率进行抽样：

$$F_s \geqslant 2f_{max} \tag{2.1}$$

其中F_s是抽样频率或抽样率。因此，如果模拟信号中的最大频率分量为4 kHz，那么，为了保留或捕捉信号中的所有信息，应该以8 kHz或者更高的抽样率进行抽样。小于抽样定理规定的抽样率进行抽样将导致频谱折叠，或者像频（image frequency）混叠进入到希望的频带内，以至于当我们要把抽样的数据转回到模拟信号时不能够恢复出原始信号。需要记住的很重要的一点是，信号有很多能量常常在感兴趣的最高频率之外或者包含噪声，信号的能量在很宽的频率范围内是不变的。例如，在电话中感兴趣的最高频率是大约3.4 kHz，而语音信号可能超过10 kHz。因此，如果我们没有将感兴趣的带宽之外的信号和噪声移去，那么将违反抽样定理。在实际应用中，让信号通过一个模拟抗混叠滤波器，可以达到移去感兴趣频带之外的信号的目的。

2.3.1.2　混叠和抽样后信号的频谱

假定我们以T（即抽样频率为$1/T$）的时间间隔对一个时域信号进行抽样，从图 2.4 中可以看出，同样一组抽样值存在另一个频率分量的原始信号。因此，对于低频分量可能取错频率分量，这就是所谓的混叠。在实际应用中，从分析混叠的影响或从找到解决混叠问题的方法的角度来考察频域的混叠问题将更具有启发性。

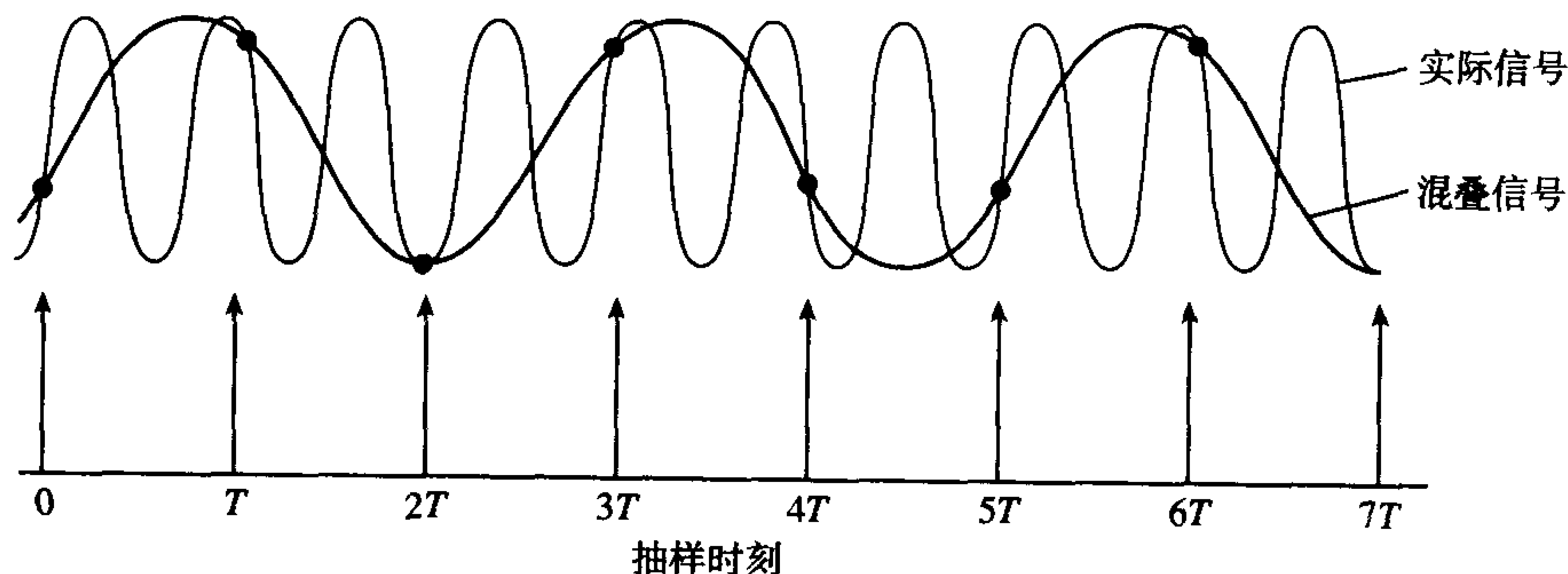


图 2.4　时域混叠的一个例子。注意，两个信号尽管它们的频率不同，但在抽样时刻有相同的值

图 2.5 显示了抽样的过程，抽样的过程可以看作为模拟信号$x(t)$乘以一个抽样函数$p(t)$。$p(t)$由一串单位幅度脉冲信号组成，脉冲的宽度为dt（无穷小量），周期为T。$x(t)$、$p(t)$以及它们相乘的谱显示在图 2.5 中。注意，$X'(f)$是$X(f)$与$P(f)$的卷积－时域相乘等价于频域卷积。

对于图 2.5(d)的抽样后的信号有几点应该引起注意：

- 频谱与原始信号的频谱相同，但是以抽样频率F_s的倍数重复，集中在以F_s倍数为中心的高阶频率分量称为像频；

● 如果抽样频率 F_s 没有足够高，那么以 F_s 为中心的像频将发生折叠，或者混叠到基带频率（参见图 2.6），在这种情况下，期望信号的信息在折叠区域将不能与像频区分出来；
● 重叠或混叠出现在点 F_N，它是抽样频率点的一半，这个频率点有很多叫法，可称为折叠频率、奈奎斯特（Nyquist）频率等。

在实际中，混叠总是会出现的，因为在感兴趣的频带外出现噪声且存在信号，那么问题是确定可接受的混叠电平，然后设计一个合适的抗混叠滤波器，并选择一个合适的抽样频率来实现它。

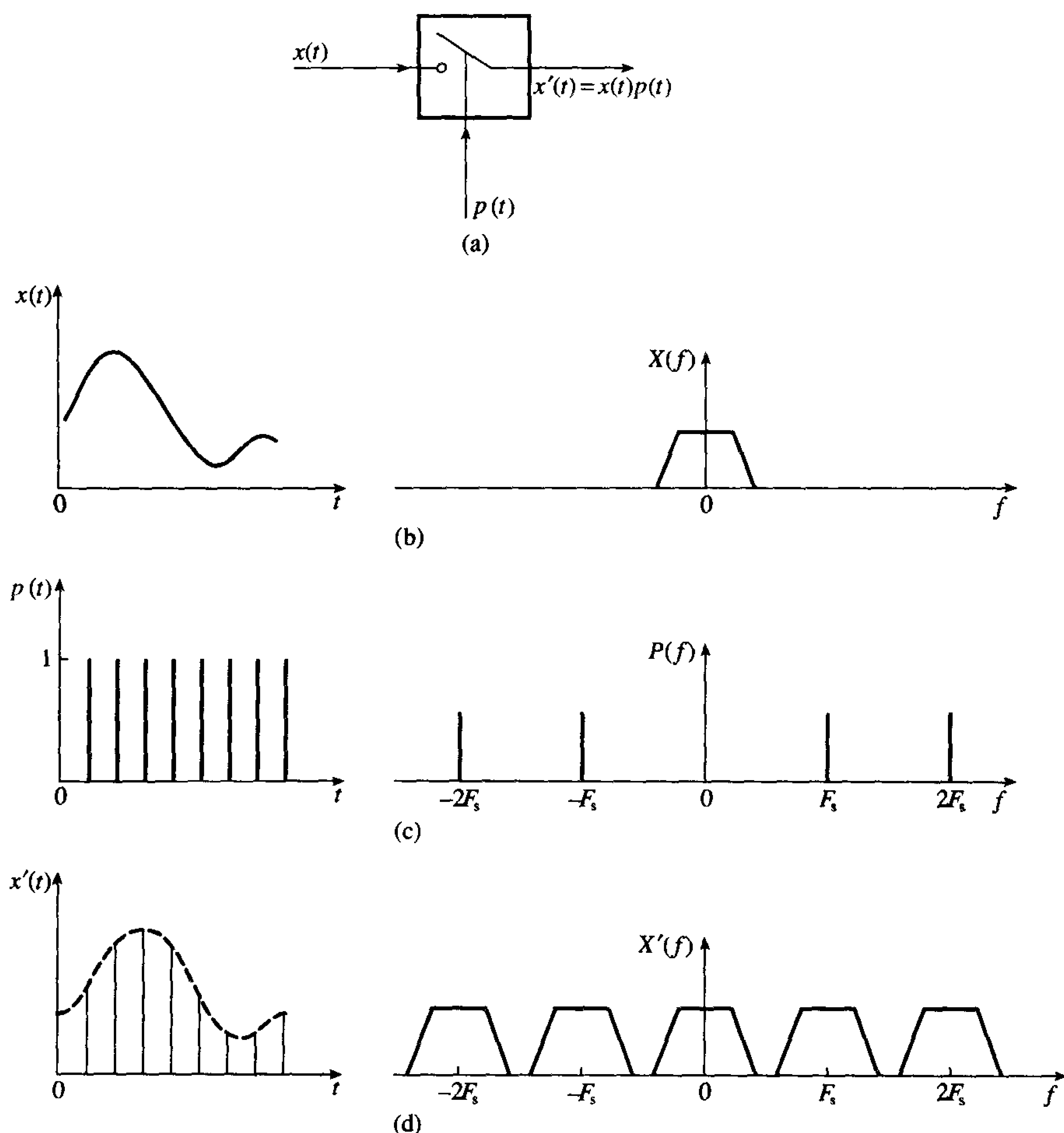


图 2.5　抽样过程的时域和频域表示，应该比较抽样前信号(b)的频谱和抽样后信号(d)的频谱。在(d)中注意抽样后信号频谱的变化，特别是抽样后信号频谱以抽样频率 F_s 的倍数重复

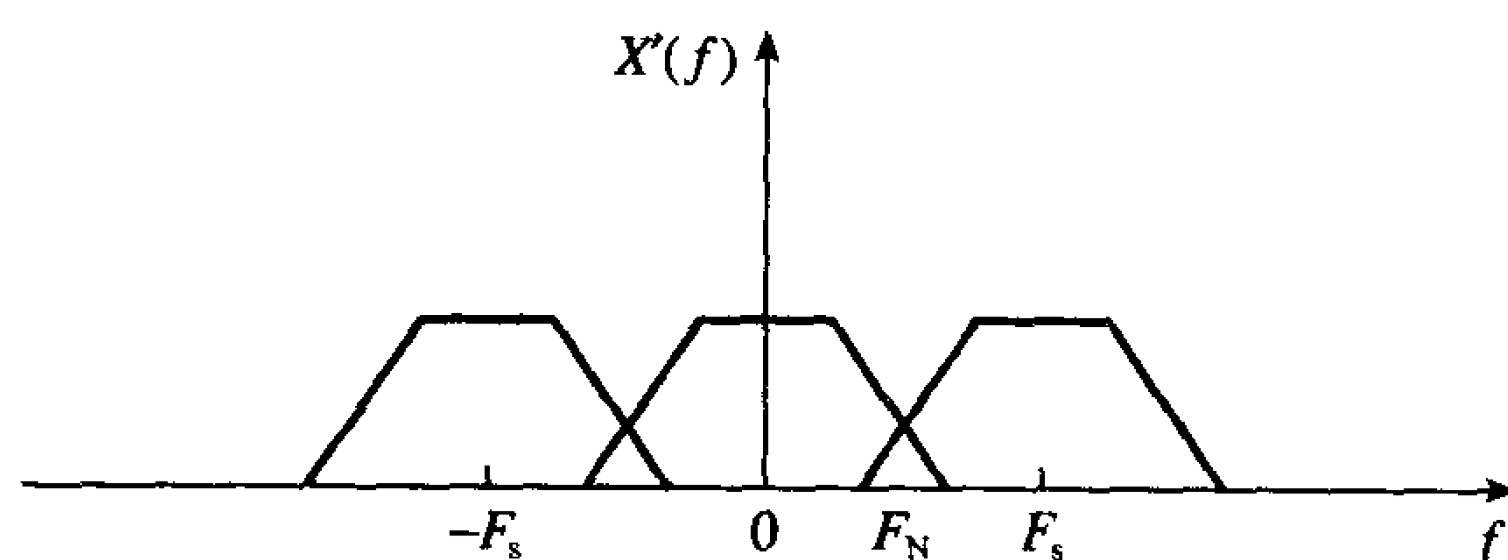


图 2.6　欠抽样信号的频谱，其中显示出了混叠（折叠区域）。折叠区域中的信号并没有恢复。F_N 等于抽样频率的一半，通常称为奈奎斯特频率。为了恢复信号的所有分量，必须在大于等于两倍最高频率分量的速率上进行抽样

2.3.1.3　抗混叠滤波器

为了减少混叠的影响，对带限信号通常采用锐截止抗混叠滤波器，或者增加抽样频率，使信号与像频频谱分得较开。理想的情况是抗混叠滤波器应该移去大于折叠频率的所有频率分量，所以理想的抗混叠滤波器应该具有类似于图2.7(a)所示的频率响应特性。图2.7(b)给出了实际的响应特性，其中f_c和f_s分别为截止频率（cutoff frequency）和阻带频率（stopband frequency）。从图2.7(b)和图2.7(c)我们注意到，实际的响应引入了信号的幅度失真，因为在通带内响应特性不是平坦的。另外，大于f_s的信号分量衰减了A_{min}；而在f_c和f_s之间的频率分量（即过渡带），它们的幅度是单调衰减的。

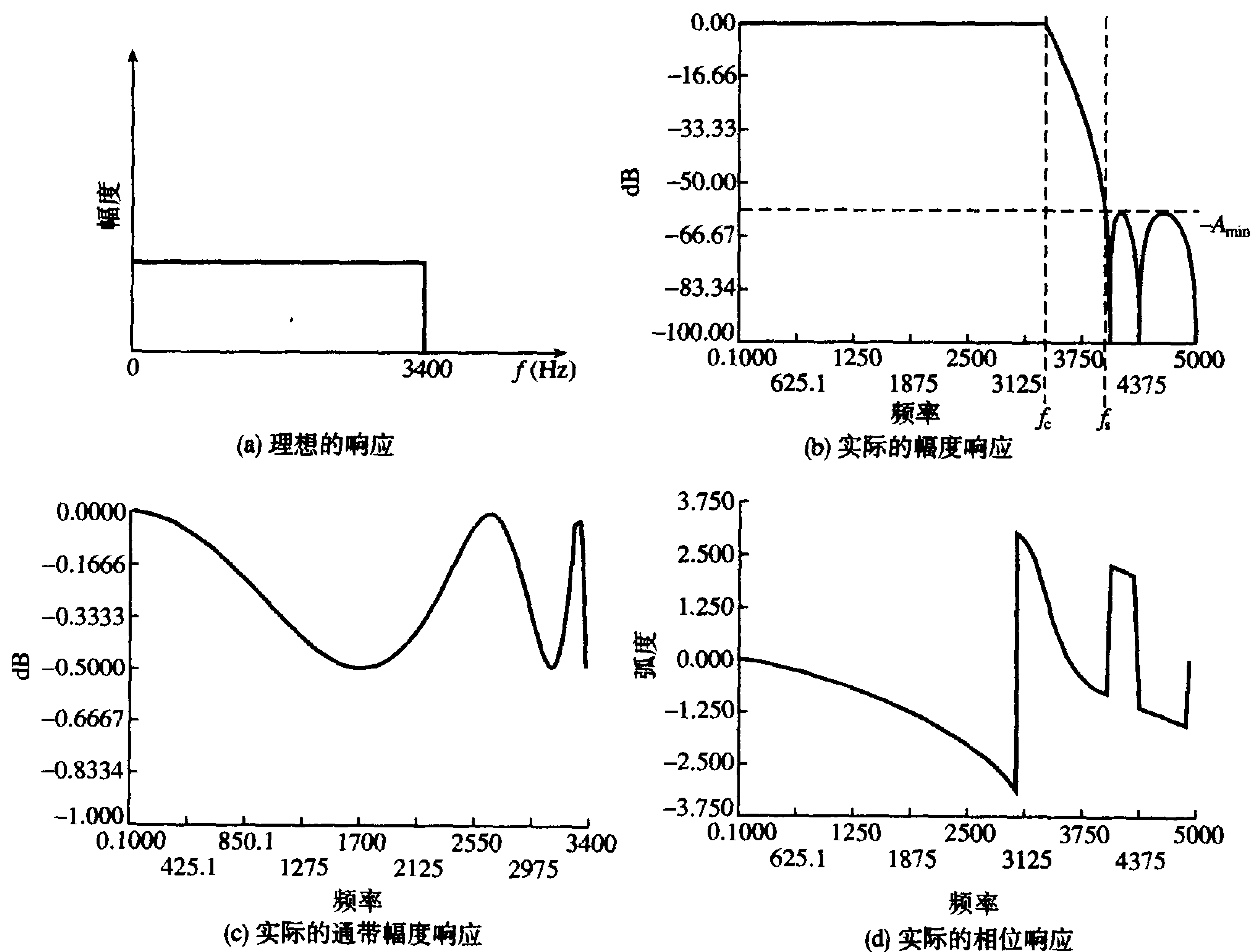


图 2.7　理想的和实际的抗混叠滤波器的频率响应，显示了由实际的响应引入的误差。例如，比较(a)中的平坦通带响应和(c)中的实际的通带响应：实际响应中的波纹将引入带内信号分量的幅度失真

抗混叠滤波器应该对奈奎斯特频率以上的频率分量提供足够的衰减。实际的滤波器由于非理想的响应特性，有效的奈奎斯特频率取为f_s（阻带边缘频率）。在确定抗混叠滤波器时，考虑ADC分辨率要求是有益的。因此，滤波器应该设计成将奈奎斯特频率以上的频率分量衰减到ADC检测不到的电平，例如小于量化噪声电平（参考后面的内容）。于是，对于采用B位线性ADC的系统，其滤波器的最小阻带衰减通常应该是

$$A_{min} = 20 \log (\sqrt{1.5} \times 2^{B}) \qquad (2.2)$$

其中B是ADC的位数（进一步的讨论请参见例2.3），表2.1给出了对不同B值的A_{min}。

表 2.1 对不同的 ADC 分辨率 B，最小低通滤波器阻带衰减 A_{min} 的估计

B	A_{min}(dB)
8	50
10	62
12	74
16	98

在 DSP 系统的前端采用模拟滤波器也会引入另外一种限制，如相位失真。图 2.7(d)画出了抗混叠滤波器的相位响应，它的幅度响应由图2.7(c)给出，这个图表明相位响应不是频率的线性函数。所以，期望的信号将产生相移，或者产生与它们频率分量不成比例的延迟量。失真程度与滤波器的特性有关，包括它的下降沿（roll-off）陡峭的程度。在许多情况下，下降沿越陡峭（即过渡带越窄），由滤波器引入的相位失真越厉害，幅度上要达到好的匹配就越困难。在多通道系统中，通道之间将产生群延迟。然而，采用陡峭下降沿滤波器允许采用较低的抽样率，以及低速的、便宜的 ADC。

实时信号处理的趋势是采用高的抽样频率，也就是对信号进行过抽样，尽管可能冒着采用快速的、昂贵的 ADC 的风险。这样做的理由是多方面的：首先，采用了使相位失真最小的简单抗混叠滤波器，对多通道系统而言成本得以降低；其次，过抽样与附加的数字信号处理一起使信噪比得到改善（参见第 9 章）。为了可用于不同的应用领域，具有模拟前端的 DSP 系统的截止频率必须是可变的。因此，开始使用可编程的模拟滤波器，例如 MF10，但是它们的性能不是完全令人满意，并且对于多通道系统可能过于昂贵。模拟信号的过抽样允许采用数字抽样率转换技术（参见第 9 章），以达到很容易实现可变截止频率的要求。

2.3.1.4 抽样频率选择和混叠控制举例

影响抽样频率选择的因素包括下列几点：

- 输入信号的频率成分
- 抗混叠滤波的要求
- 可接受的混叠误差电平
- ADC 的分辨率
- 存储的要求

我们已经讨论了前面两点对抽样频率的影响，有许多方法能够规定可接受的混叠误差电平。在大多数情况下，需要利用抽样频率、混叠误差电平和滤波器参数之间的关系。例如，对于给定的抗混叠滤波器的特性，我们可以规定可接受的混叠误差电平，然后确定达到混叠误差电平所必需的抽样频率。或者对于给定的抽样频率，规定混叠误差电平，我们可以计算出由抗混叠滤波器提供的最小阻带衰减。在实际中，应该考虑 ADC 分辨率，因为它建立了系统噪声的基数。

下面的例子说明了选择抽样频率和混叠控制所涉及到的有关问题。

例 2.1 说明抽样的影响以及混叠误差和抽样频率之间的内部关系 实时 DSP 系统的前端在图 2.8 中给出，假设一个宽带输入信号。

(a) 画出抽样之前（A 点）和抽样之后（B 点）信号在 $\pm F_s/2$ 之间的频谱；

(b) 确定在 10 kHz 与奈奎斯特频率（即 20 kHz）处信号和混叠误差电平；

(c) 给定 10 kHz 处信号－混叠误差电平比为 10 : 1，确定最小抽样频率 F_s(min)，阐述做出的其他假定。

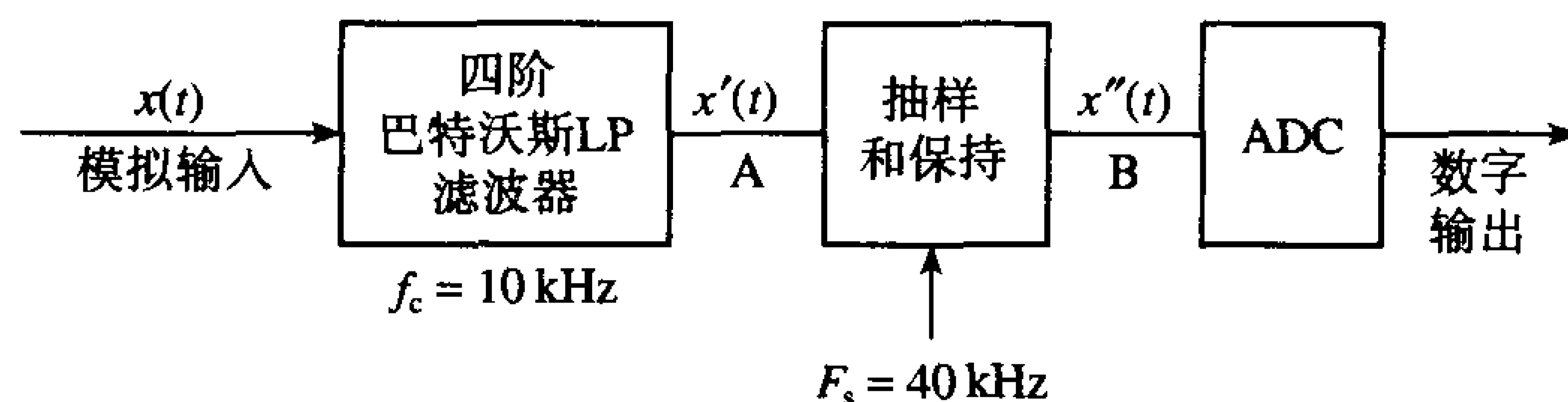


图 2.8　实时 DSP 系统的前端

解:

(a) 抽样前后信号的频谱绘于图 2.9 中，我们注意到每个频谱分量的形状由巴特沃斯（Butterworth）滤波器的响应等式确定，巴特沃斯滤波器的响应为

$$|H(f)| = \frac{1}{\sqrt{1+\left(\frac{f}{f_c}\right)^8}}$$

(b) 在滤波器输出端信号的频谱等于信号频谱与滤波器响应的乘积，即 $X(f)|H(f)|$。对于宽带输入，频谱 $X(f)$基本上是平坦的，如果我们假定 $X(f)$和 $H(f)$有最大值 1（即归一化），那么抽样前后的信号电平由模拟滤波器的形状控制。

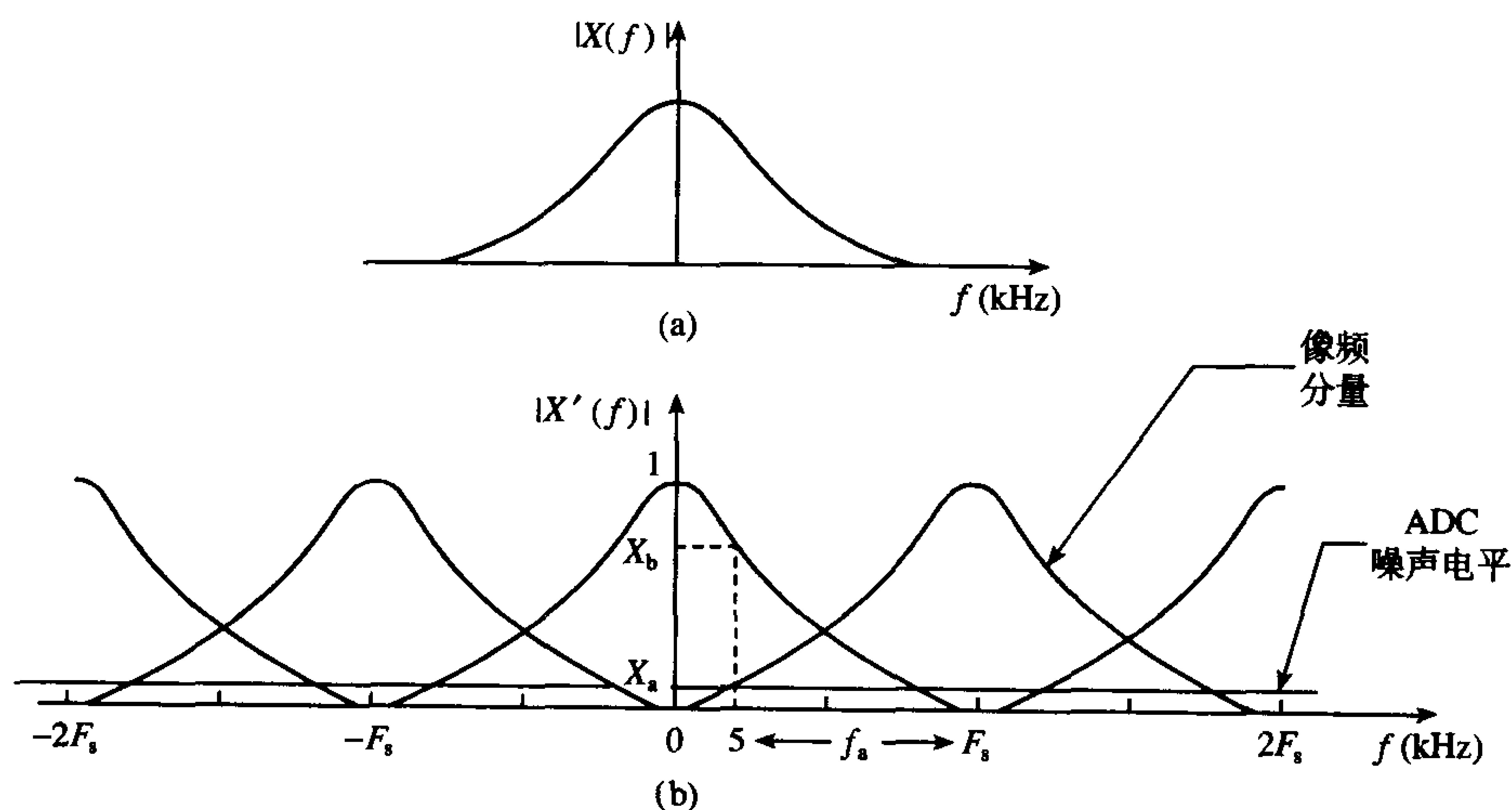


图 2.9　抽样信号的频谱表明由于混叠和 ADC 量化引起的误差

因此，在 10 kHz 处和f_c= 10 kHz，归一化的信号电平（由前面的等式）为 0.707（即 $1/\sqrt{2}$），混叠误差电平（由图 2.9(b)得出）由下式给出：

$$\text{混叠电平，} X_a = \frac{1}{\sqrt{1+\left(\frac{30}{10}\right)^8}} = 0.012$$

奈奎斯特频率是 20 kHz（即抽样频率的一半），这是图 2.9(b)中的交叉点，所以信号与混叠误差电平是相同的。在 20 kHz 处信号和混叠电平每个都等于 0.062（采用巴特沃斯等式，f = 20 kHz，f_c = 10 kHz）。

(c) 在 10 kHz 处，信号电平是 0.707，信号电平与混叠电平之比是 10 : 1，这意味着混叠电平是 0.0707。引起混叠的像频分量由巴特沃斯方程控制。

这样，由

$$\frac{1}{\sqrt{1+\left(\frac{f}{10}\right)^{8}}}=0.0707$$

我们求得f=19.39 kHz。

这个频率对应于10 kHz处的混叠频率，即上面图2.9(b)中的f_a。因此，抽样频率为$F_s=f_a+10=29.39$ kHz。

例2.2 图2.10描述了简单数据采集系统的前端，给定混叠误差小于通带内信号电平的2%，确定最小抽样频率F_s。

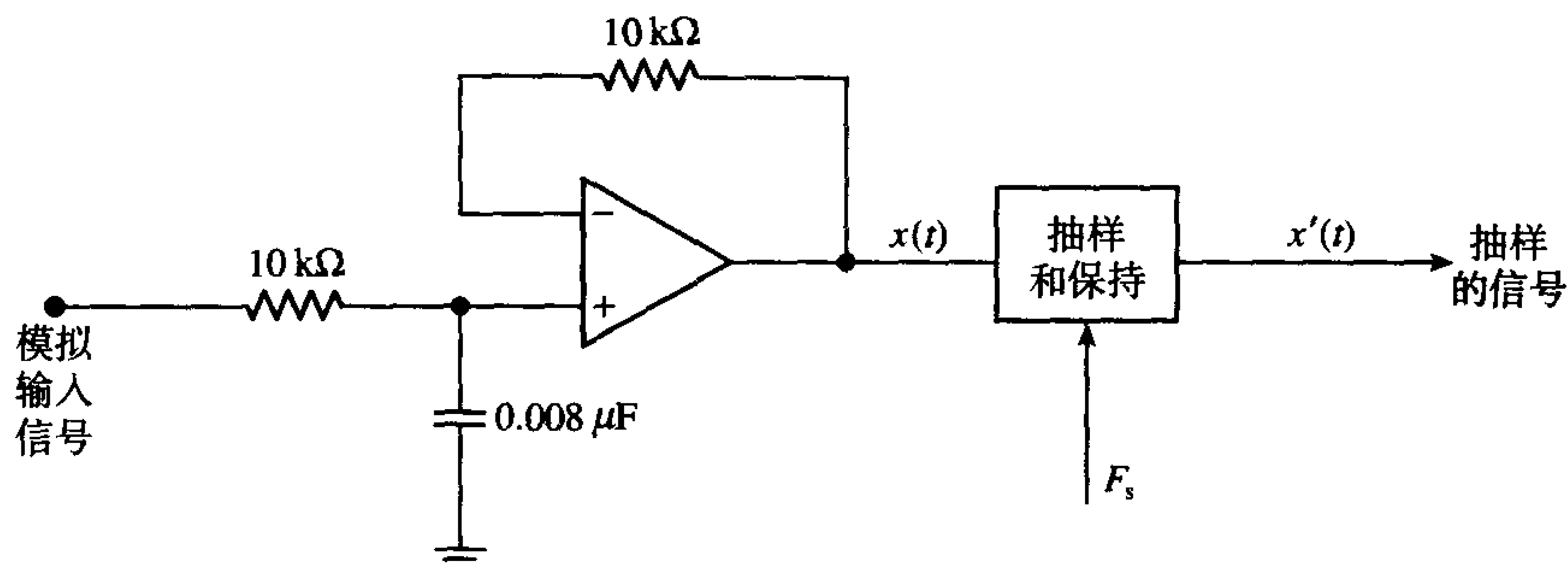


图2.10 简单数据采集系统的前端，简单的有源滤波器用来在信号以F_s速率被抽样之前限制信号的频率

解：

有源滤波器的幅度响应为

$$|H(f)|=\frac{1}{[1+(f/f_c)^2]^{1/2}} \quad 其中\ f_c=1/2\pi RC=2\ \text{kHz}$$

图2.11画出了对输入信号限制频带后的频谱和抽样信号的频谱，其中，我们假定模拟输入信号是宽带的。

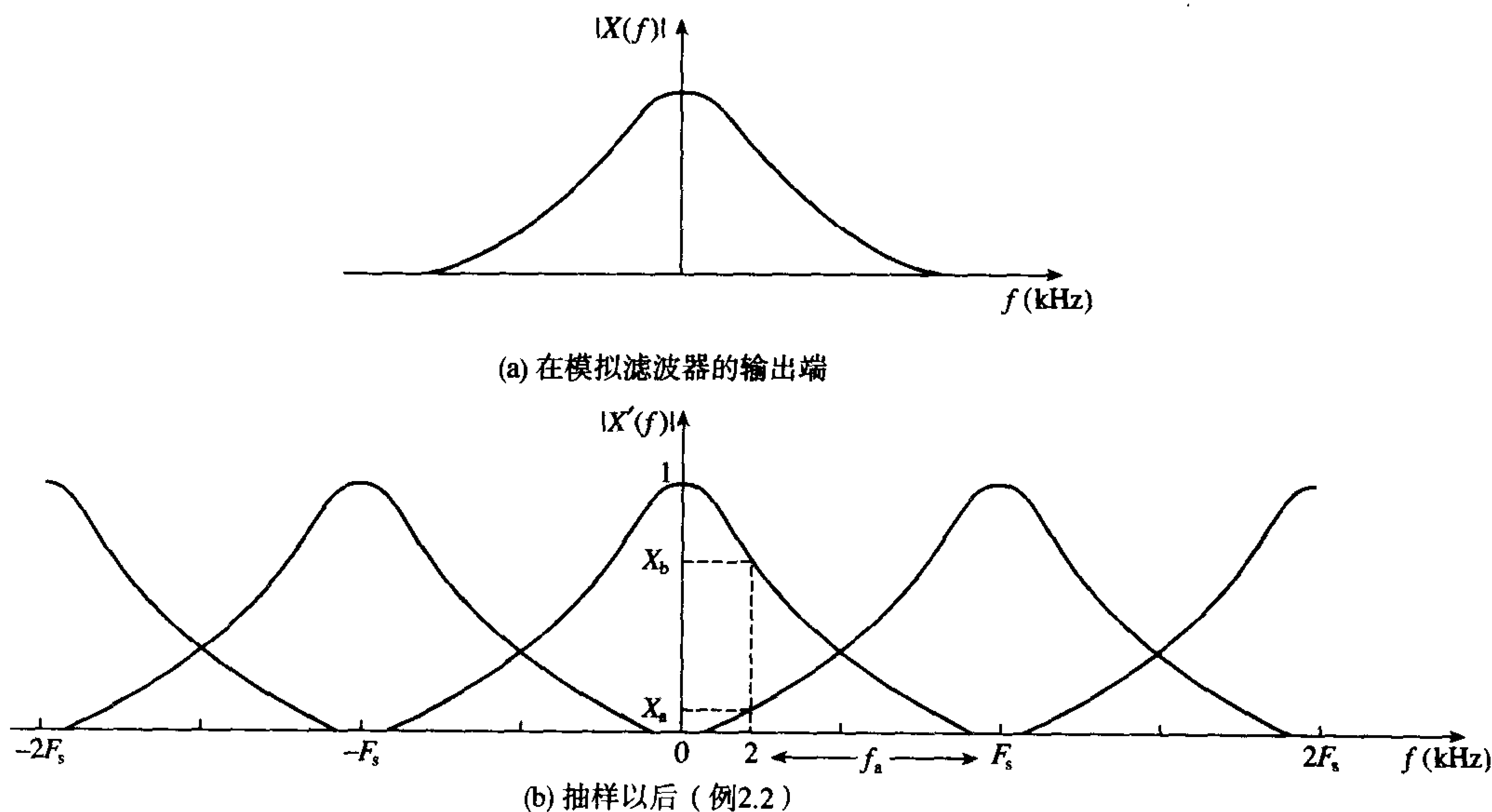


图2.11 信号的频谱

从图中我们注意到，抽样信号的频谱以抽样频率的倍数重复出现，像频折叠到期望的频带内（0～2 kHz）就是混叠的频谱。

在 2 kHz 处，信号电平为 $X_b = 0.7071$，所以

$$\text{期望的混叠电平} < 0.7071 \times 2/100 = 0.014\,14$$

因此

$$0.014\,14 < \frac{1}{[1 + (f_a/2)^2]^{1/2}}$$

其中 f_a 是混叠频率，求解 f_a 我们有 f_a<141.4 kHz，因此

$$F_s(\min) > f_c + f_a = 2\text{ kHz} + 141.4\text{ kHz} = 143.4\text{ kHz}$$

为了满足要求，并且考虑在 $2F_s$、$3F_s$（忽略以上的像频）为中心的像频的影响，那么 $F_s(\min)$ >143.4 kHz。令 $F_s(\min) = 150$ kHz。

例 2.3　举例说明 ADC 分辨率与滤波器参数之间的内部关系　图 2.12 给出了一个实际的实时 DSP 系统，假定感兴趣的带宽扩展到 0～4 kHz，采用 12 位双极性 ADC，估算：

(1) 抗混叠滤波器的最小阻带衰减 $A_{\min}$；

(2) 最小抽样频率 F_s；

(3) 对于估算的 $A_{\min}$ 和 F_s，相对于阻带内的信号电平的混叠误差。

假定输入的是一个宽带信号，画出并标记模拟滤波器输出端以及抽样后信号的频谱。

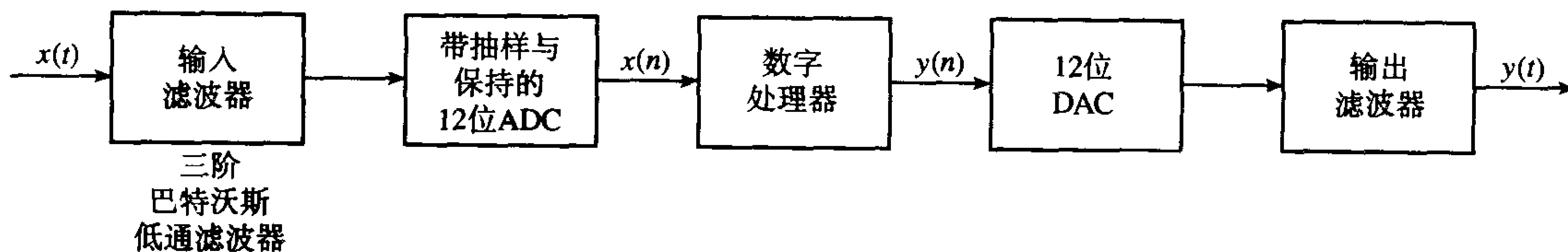


图 2.12　实时数字信号处理系统

解：

为了满足抽样定理，抗混叠滤波器带限输入信号频谱，使得在奈奎斯特频率以上的频谱被滤除，避免混叠。

在实际中，由于我们没有理想的滤波器，抗混叠滤波器通常都要求衰减奈奎斯特频率以上的小于 ADC 量化噪声电平的均方根（RMS）的频率分量，以便使 ADC 检测不到它们。

图 2.13 画出了抗混叠滤波器典型的幅度频率响应，并且显示了通带、过渡带和阻带。设计抗混叠滤波器是为了使阻带内（即频率≥$f'_{\max}$）频率分量的电平衰减到小于 DAC 量化噪声电平的均方根。

因此，有效的奈奎斯特频率是 $f'_{\max}$，有效的抽样率定义为

$$F_s \geqslant 2f'_{\max} \tag{2.3}$$

量化步长尺寸 q 由下式给出：

$$q = \frac{V_{fs}}{2^B - 1} \approx \frac{V_{fs}}{2^B}$$

其中 B 是 ADC 的位数，V_{fs} 是满刻度输入范围，因此

$$V_{fs} \approx q \times 2^B$$

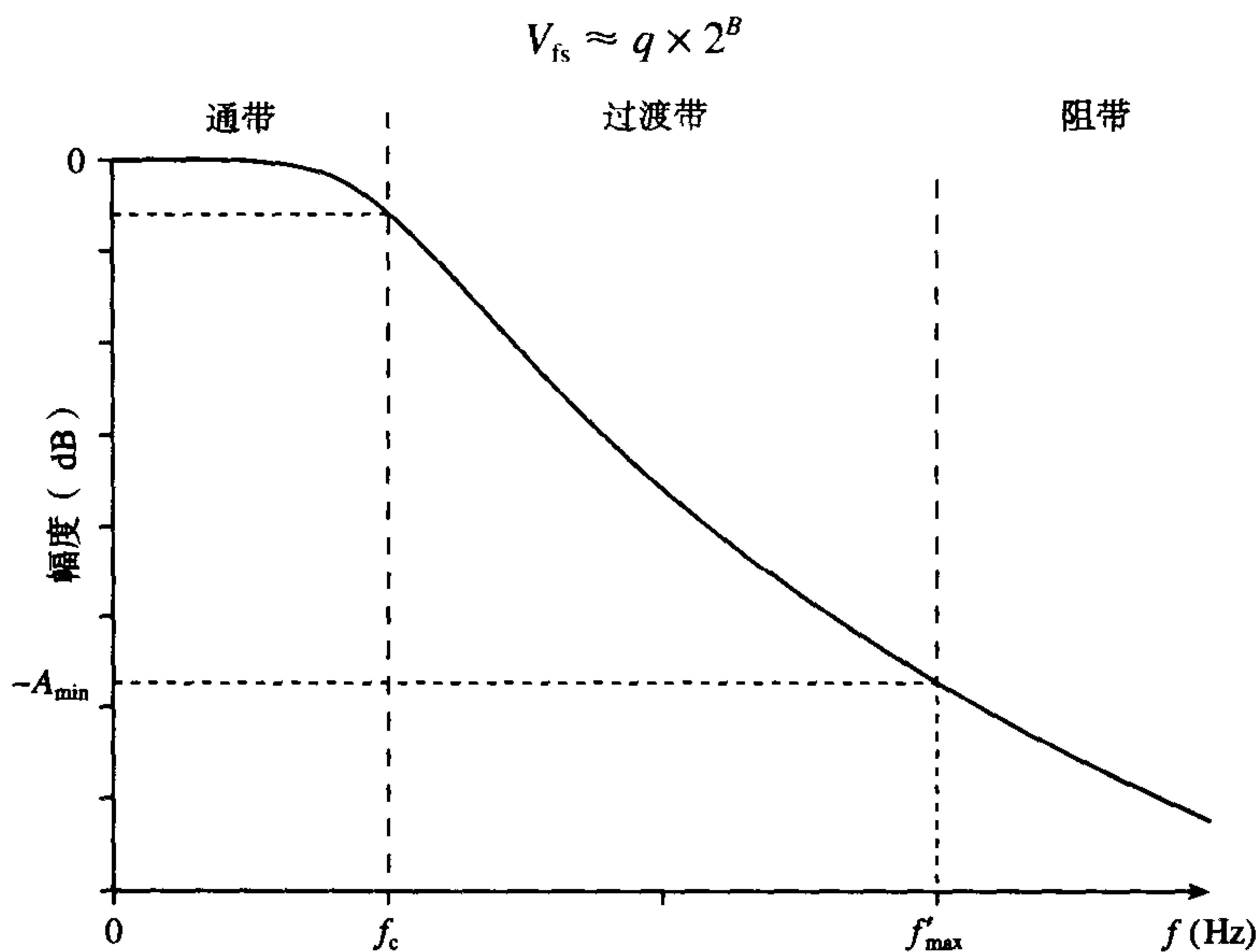


图 2.13　实际的抗混叠滤波器的典型幅度频率响应

量化噪声电平的均方根为

$$\sqrt{\frac{q^2}{12}} = \frac{q}{2\sqrt{3}}$$

为了简单起见，如果我们假定输入是幅度为 A（刚好填满 ADC 范围）的正弦波，那么最大的通带信号电平是

$$V_{fs} = 2A = q \times 2^B$$

因此，

$$A = \frac{q \times 2^B}{2}$$

最大通带信号电平与阻带信号电平之比给出了滤波器最大阻带衰减的度量：

$$\frac{\text{最大通带信号电平}}{\text{阻带信号电平}} = \frac{q \times 2^B/2\sqrt{2}}{q/2\sqrt{3}}$$

$$= \sqrt{1.5} \times 2^B$$

(1) 因此，对于 DSP 系统，最小阻带衰减 A_{min} 由下式给出（对正弦波输入）:

$$A_{min} = 20\log(\sqrt{1.5} \times 2^B)\ \text{dB}$$

$$= 74\ \text{dB}$$

(2) 抽样前后信号的频谱（忽略高阶像频）如图 2.14 所示。

由

$$A_{min} = 74 = 20\log\left[1 + \left(\frac{f'_{max}}{f_c}\right)^6\right]^{\frac{1}{2}}$$

我们得到

$$\left(\frac{f'_{max}}{f_c}\right)^6 = (5011.87)^2 - 1$$

因此，f'_{max} 等于68.45 kHz。

由 2.3 式，我们有 $F_s = 2f'_{max} = 136.9$ kHz。

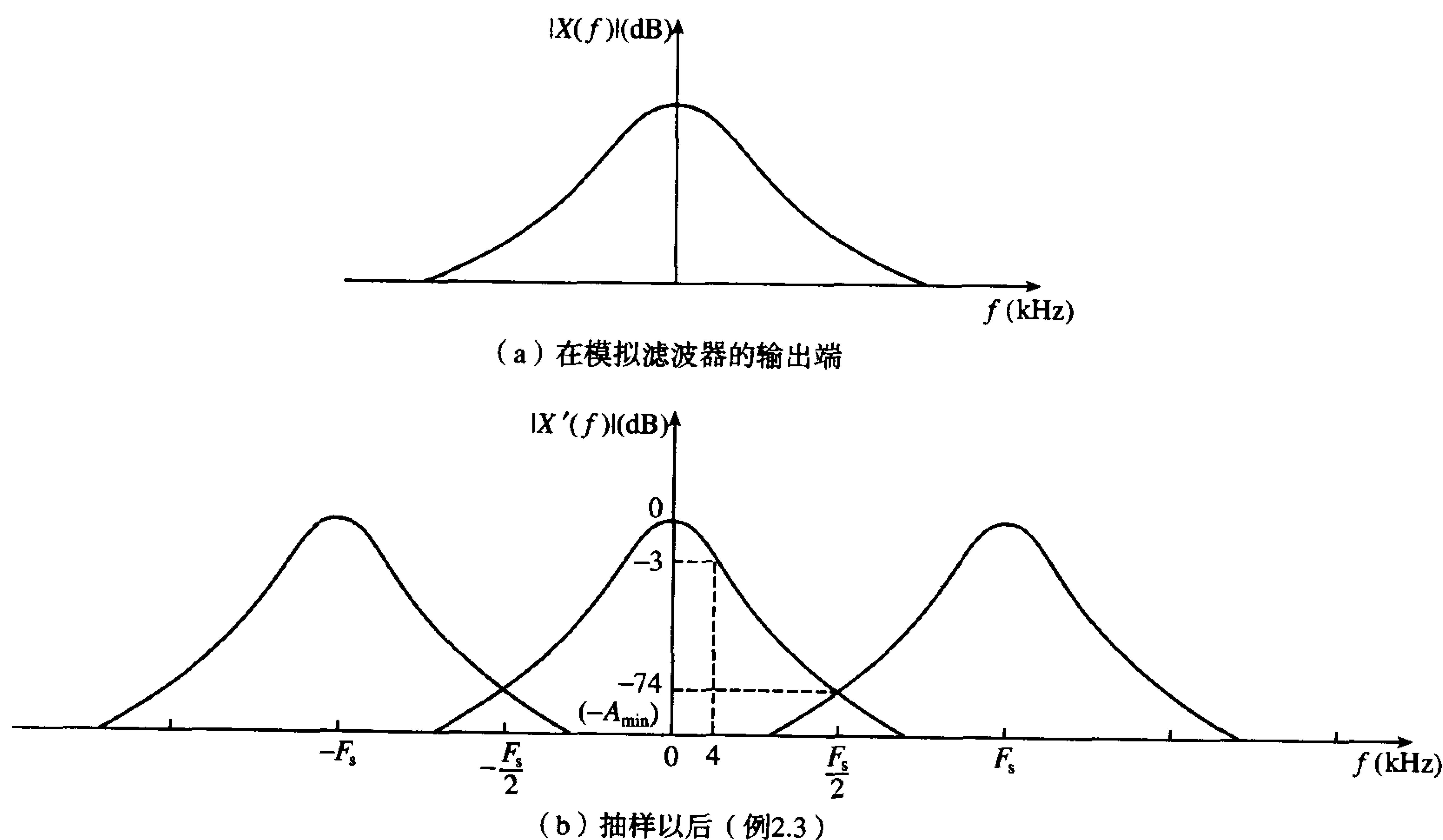
(3) 在 4 kHz 处的混叠电平为

$$\frac{1}{\left[1+\left(\frac{136.9-4}{4}\right)^6\right]^{\frac{1}{2}}} = 2.73\times10^{-5}$$

在 4 kHz 处，相对于信号电平的混叠电平为

$$\frac{2.73\times10^{-5}}{0.7071}$$

如果我们希望带沿频率刚好衰减到量化噪声电平以下，那么抽样频率可以减少为68.4 kHz+4 kHz = 72.4 kHz。



（a）在模拟滤波器的输出端

（b）抽样以后（例2.3）

图 2.14　信号的频谱

例 2.4　用 ADC 噪声水平作为参考　具有均匀功率谱密度的模拟信号被具有下列幅度响应的抗混叠滤波器所限带：

$$|H(f)| = \frac{1}{\left[1+\left(\frac{f}{f_c}\right)^8\right]^{\frac{1}{2}}}$$

其中 $f_c = 5$ kHz，信号用 12 位线性双极性 ADC 数字化，确定：

(1) 使通带内最大混叠误差不大于量化误差电平的最小抽样频率；

(2) 相对于 ADC 量化噪声水平的最大通带信号电平（用 dB 表示）。

解：

(1) 选择抽样频率，使得抗混叠滤波器将折叠到通带内的混叠噪声衰减到小于ADC的最大均方根量化电平，以至于ADC检测不到它们（参见图2.15）。

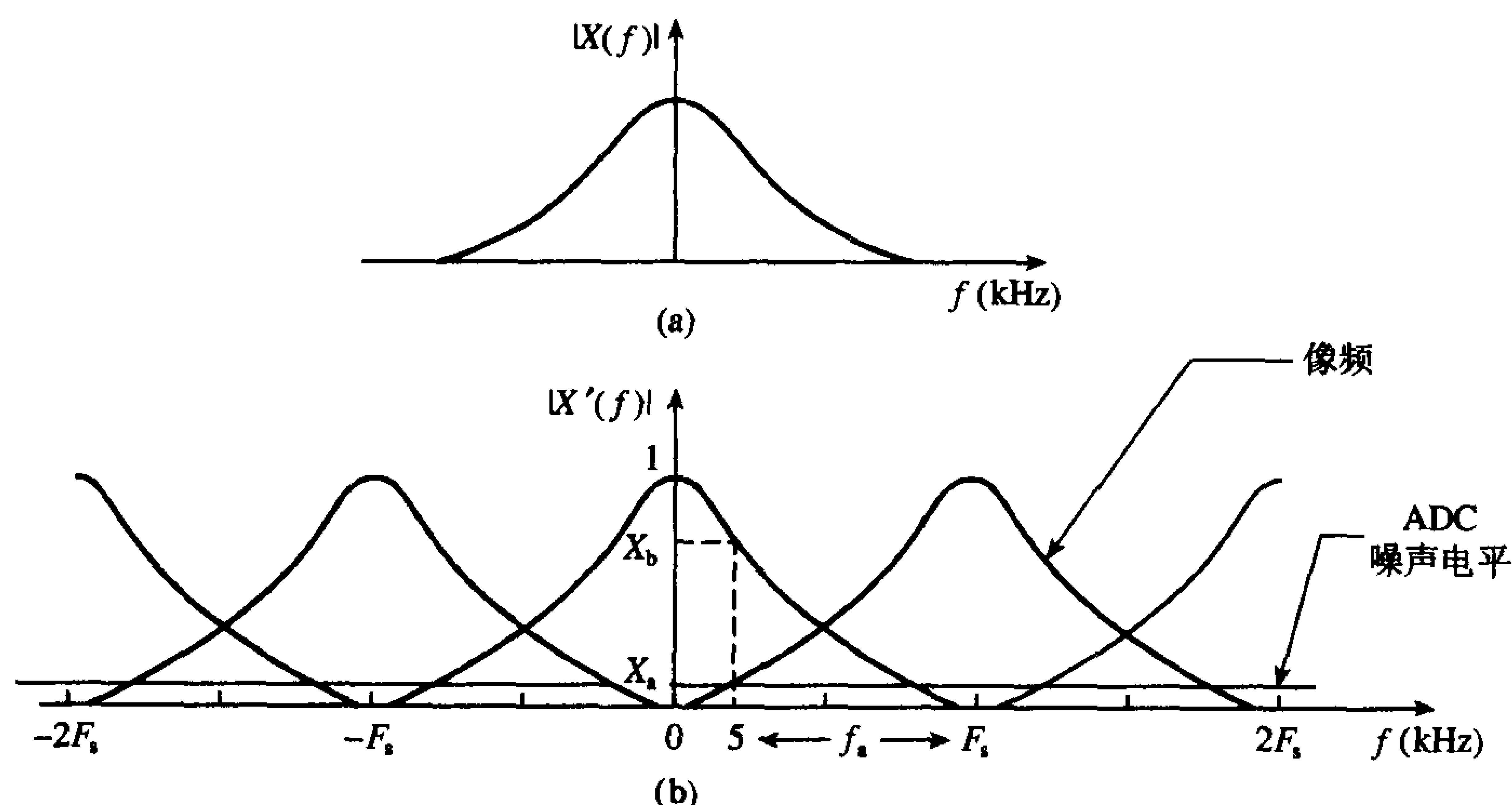


图2.15 抽样后信号的频谱表明由于混叠和ADC量化（例2.4）引起的误差

假定输入是幅度为A（刚好填满ADC的输入范围）的正弦，那么

$$\text{输入信号的均方根：}\frac{A}{\sqrt{2}}$$

$$\text{量化步长尺寸：}q \approx \frac{2A}{2^B}$$

$$\text{量化噪声的均方根：}\frac{q}{2\sqrt{3}} = \frac{A}{\sqrt{3} \times 2^B}$$

在5 kHz处，折叠的最大混叠噪声：

$$\frac{A}{\sqrt{2}} \times \frac{1}{\left[1+\left(\frac{f_a}{5}\right)^6\right]^{\frac{1}{2}}} = \frac{A}{\sqrt{3} \times 2^B} \tag{2.4}$$

取$B = 12$位，我们可以解得混叠频率f_a以及抽样频率F_s为

$$f_a = 85.59\ \text{kHz}$$

$$F_s = f_a + 5 = 90.59\ \text{kHz}$$

(2) 相对于ADC噪声水平的最大信号：

$$\frac{\text{最大均方根信号电平}}{\text{最大量化误差}} = \frac{A/\sqrt{2}}{A/(\sqrt{3} \times 2^B)} = \sqrt{1.5} \times 2^B$$

$$\text{信号：ADC噪声水平} = 20\log(\sqrt{1.5} \times 2^B) \tag{2.5}$$

另外注意，在这种情况下信号与ADC噪声水平之比可以从下式得到（参见2.4式）：

$$\text{信号：噪声水平} = 20\log\left[1+\left(\frac{f_a}{5}\right)^6\right]^{\frac{1}{2}}\ \text{dB}$$

2.3.1.5　其他与抽样有关的实际问题：精度和带宽限制

在实际的系统中，图 2.5(d)所示的瞬时抽样是不可能的，抽样函数是有限宽度的，这带来了所谓的孔径效应（aperture effect）的问题，表示信号是在一个有限的时间间隔上而不是瞬时测到的。非零孔径时间限制了精度和能够被数字化的最大信号频率，因为抽样时信号可能改变。我们假定如果在孔径间隔时间内信号最大的变化为$\frac{1}{2}$ LSB（最低有效位），那么就可以得到孔径效应的度量。因此，对于一个采用 B 位 ADC 的系统，当输入为正弦波时，能够以$\frac{1}{2}$ LSB 数字化的最大频率为

$$f_{\max}=\frac{1}{\pi 2^{B+1}\tau} \tag{2.6}$$

其中 τ 是孔径时间（证明参见例 2.5）。

例 2.5　一个 12 位 ADC、转换时间为 35 μs、无抽样保持的实时 DSP 系统。以$\frac{1}{2}$ LSB 精度数字化的最高频率是多少？假定二进制系统采用均匀量化，解释结果。

解：

考虑一个正弦波信号，它的峰值幅度等于 ADC 满刻度范围的一半，即 $V_{fs}/2$（参见图 2.16），图中 τ 是孔径时间，Δv 是 $v(t)$ 在间隔 τ 期间的变化，最大的变化点在 $t=0$。为了以期望的精度来测量信号，ADC 必须处理最大变化点，在这一点：

$$\left.\frac{\mathrm{d}v(t)}{\mathrm{d}t}\right|_{t=0}=(V_{fs}/2)w\cos wt=\pi f V_{fs}\quad(\mathrm{V\ s^{-1}})=\frac{\Delta v}{\tau}$$

对于$\frac{1}{2}$ LSB 精度，$\Delta v=\alpha/2$，其中 $\alpha=(V_{fs}/2^B)$，因此，$\Delta v/\tau=\pi f V_{fs}$，代入 V_{fs} 和 Δv 经化简后得

$$f_{\max}=\frac{1}{\pi 2^{B+1}\tau}$$

对于 DSP 系统，$B=12$，$\tau=35\ \mu s$，因此 $f_{\max}=1.11$ Hz。

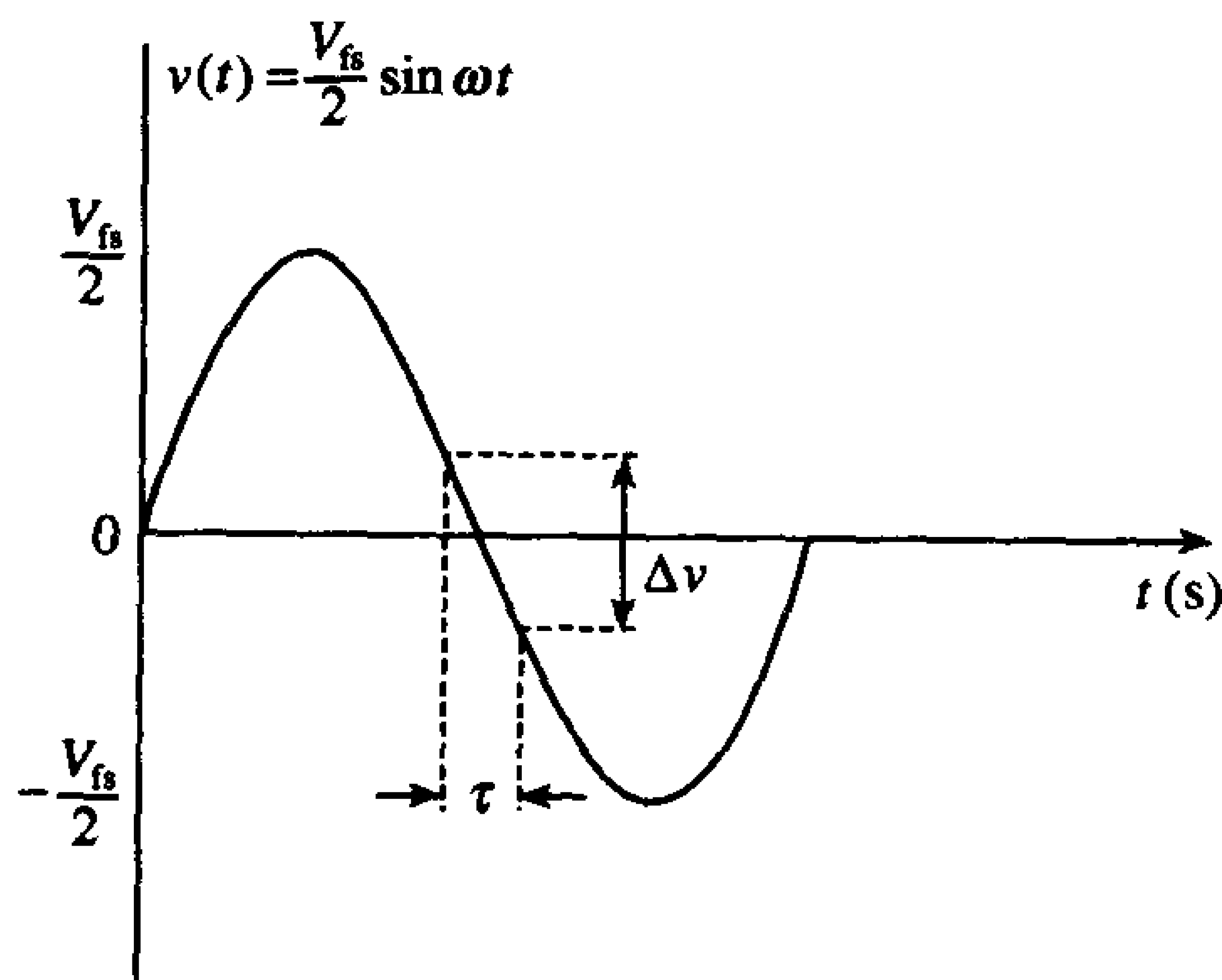


图 2.16　例 2.5 的正弦波信号

很显然，一个只能转换最大频率为 1.11 Hz 的 ADC 是没什么用处的。在实际中，ADC 前常常加有一个抽样保持电路，用来在转换期间冻结信号抽样值，使得在千赫兹范围内的信号能够精确地被数字化。例如，如果上面的 ADC 加上一个孔径时间为 25 ns 的抽样保持电路，且采集时间为 2 μs，那么能够转换的最大频率变成

$$2f_{\max}\leqslant F_s=1/(35+2+0.025)\times 10^{-6}\ \mathrm{kHz},\quad f_{\max}=13.5\ \mathrm{kHz}$$

因此，最大频率为 13.5 kHz 的信号可以以 27 kHz 的速率或者以(35 + 2 + 0.025) μs = 37.025 μs 的间隔进行抽样。

2.3.2 带通信号的抽样

2.3.2.1 引言和基本原理

在某些应用中，如通信系统，感兴趣的信号只占可用频带的很窄的一部分，如图 2.17 所示。

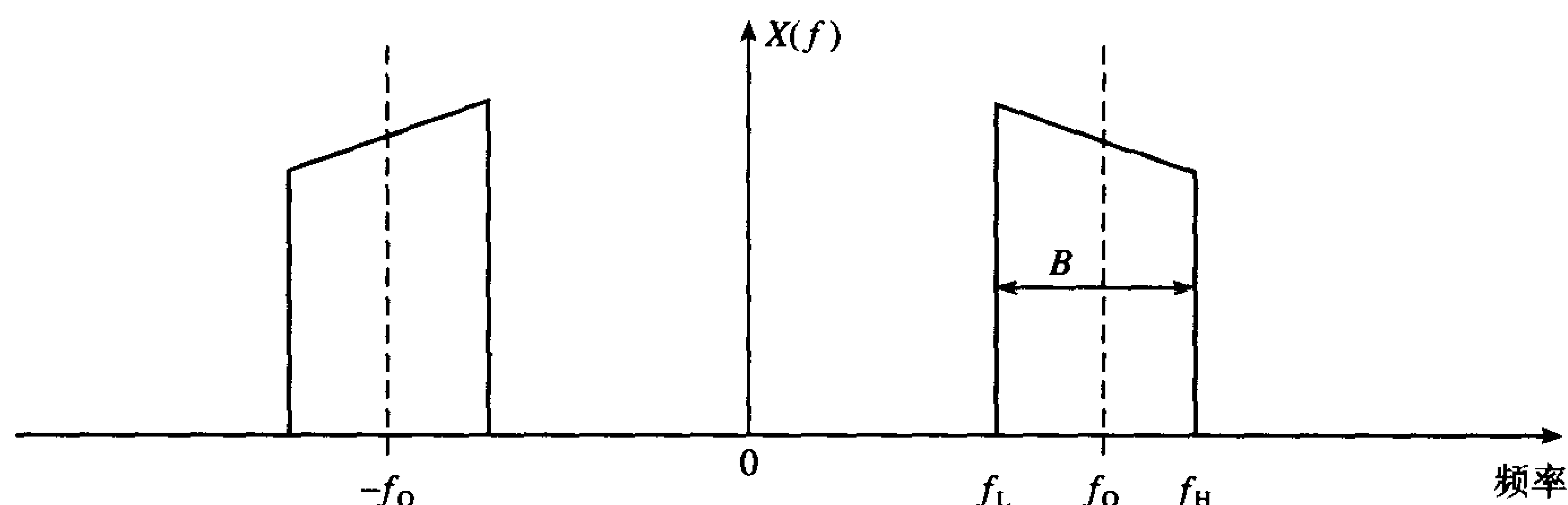


图 2.17 带通信号

在这种情况下，信号的带宽 B 与下沿和上沿频率（f_L 和 f_H）相比要小很多，因此使用低通抽样定理是不够经济的，处理这种情况的一种方法就是采用带通抽样定理（2.7 式）：

$$\frac{2f_H}{n} \leqslant F_S \leqslant \frac{2f_L}{n-1} \tag{2.7}$$

其中

$$n = \frac{f_H}{B} \quad （n 是整数，上舍入到最大的整数）$$

带通信号定理允许我们对窄带HF信号以相当低的速率抽样并且仍能避免混叠（Vaughan et al., 1991; Del Re, 1978）。带通信号的无混叠欠抽样（undersampling）有两种常用的方法，一种是所谓的整数带（integer-band）抽样，另一种采用了正交调制技术，第二种方法超出了本书的讨论范围。

2.3.2.2 整数带欠抽样技术

给定一个带通信号，如果带沿频率 f_L 和 f_H 是信号带宽的整数倍，那么就可以按理论上的最小抽样率 $2B$ 对信号抽样而不产生混叠：

$$F_s(\min) = 2B \tag{2.8a}$$

当信号的下沿频率与带宽之比或者上沿频率与带宽之比为整数时：

$$n = \frac{f_H}{B} \text{ 或者 } n = \frac{f_L}{B} \tag{2.8b}$$

2.8a 式是有效的。当 2.8b 式的条件满足时，那么称信号带宽是整数倍的。如果信号带宽不是整数倍的，那么可以扩展带沿频率，使之有效带宽变成整数倍的。

例 2.6 带通欠抽样原理性示例 多信道通信系统的前端如图 2.18(a)所示，接收信号的频谱如图 2.18(b)所示，图中标明了信道号，在信号以尽可能低的速率数字化之前，利用带通滤波器在希望的信道分离信号。

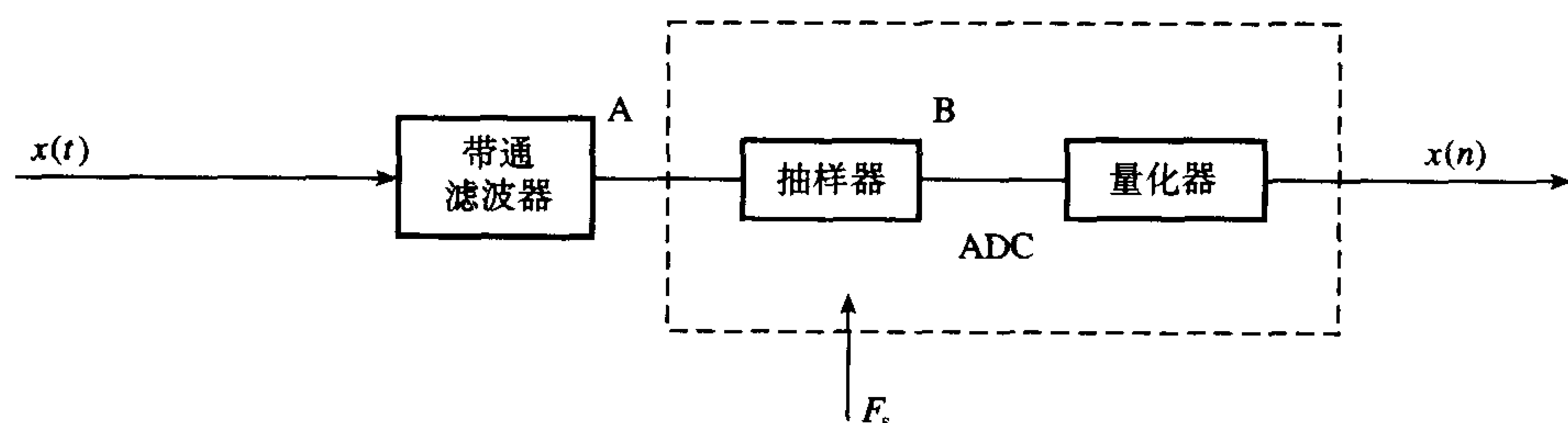
假定一个具有下列特征的理想带通滤波器：

$$H(f)=\begin{cases}1 & 40\ \text{kHz} \leqslant f \leqslant 50\ \text{kHz} \\ 0 & \text{其他}\end{cases}$$

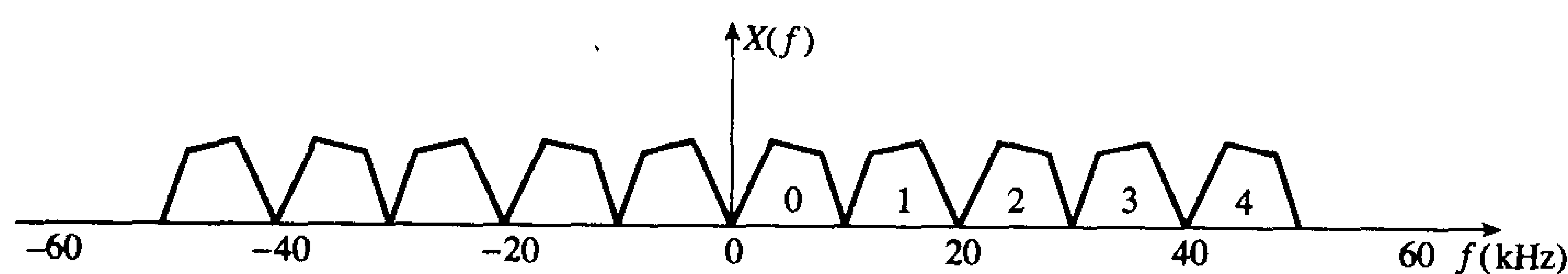
(a) (i) 确定最小理论抽样频率，

(ii) 画出抽样前（A 点）后（B 点）信号的频谱。

(b) 对于通过 3 号信道的带通滤波器重复(i)和(ii)。



(a) 系统的前端



(b) 接收信号的频谱（例2.6）

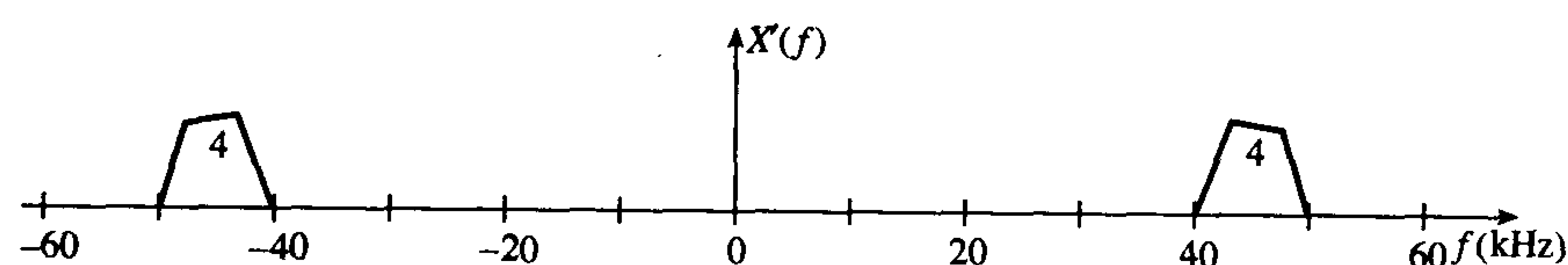
图 2.18　接收信号的频谱

解：

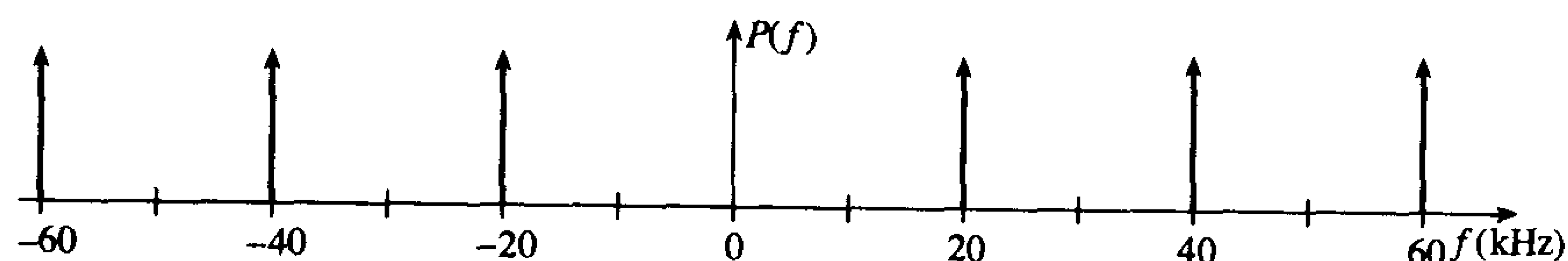
(a) (i) 最小理论抽样频率是 2 × 10 kHz，即 20 kHz。

(ii) A 点的频谱就是 4 号信道信号的频谱（参见图 2.19(a)）。

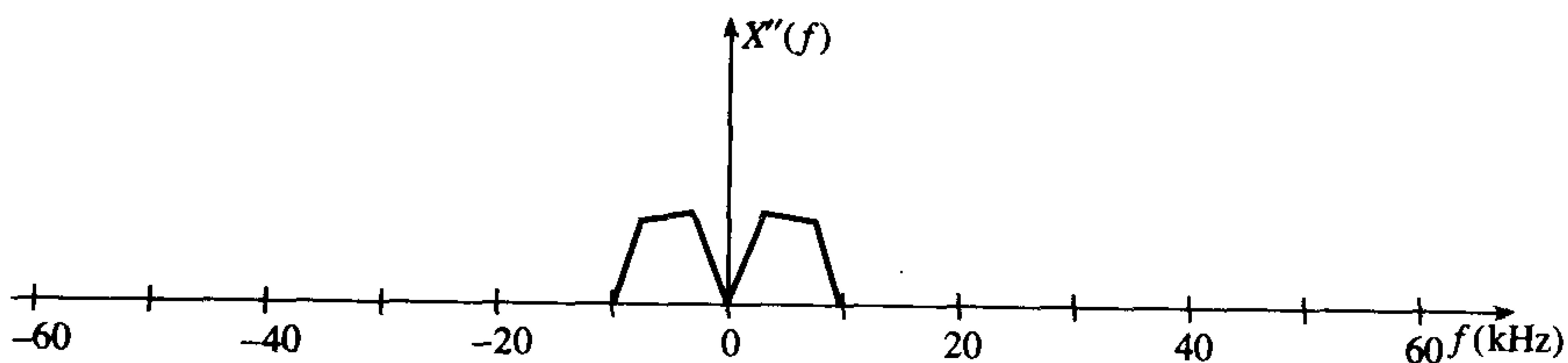
B 点（即抽样后）的频谱可以通过将带通滤波器输出信号的频谱与抽样函数的频谱进行卷积而得到（参见图 2.19(b)），结果如图 2.19(c)所示。



(a) 带通滤波器的输出



(b) 抽样函数



(c) 抽样器的输出（例2.6）

图 2.19　卷积结果

(b) (i) 抽样频率保持为 20 kHz。

(ii) 接着(a)部分，A点的频谱以及B点的频谱分别如图 2.20(a)和图 2.20(c)所示。

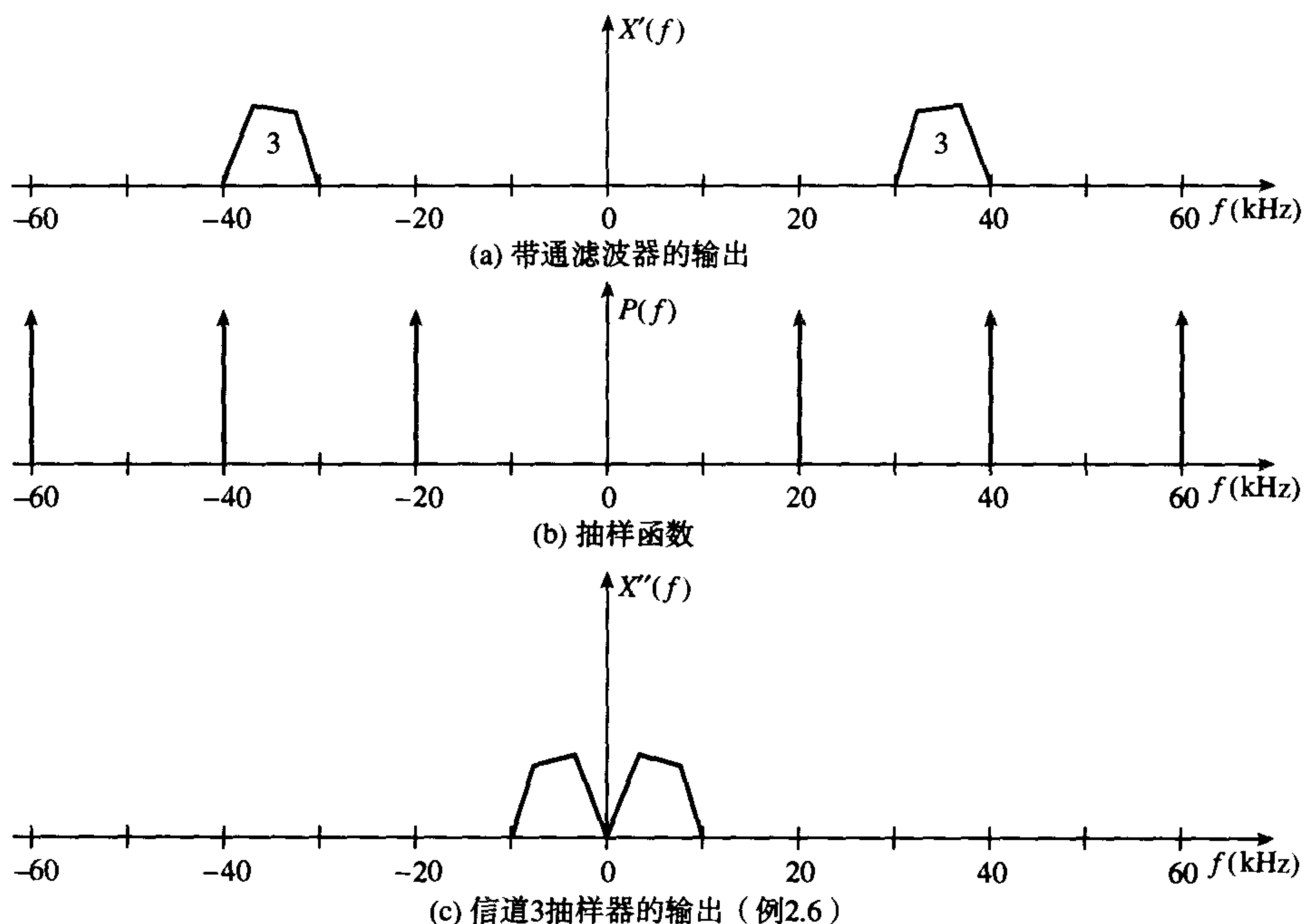


图 2.20 A点及B点的频谱

例2.7 举例说明无混叠带通欠抽样技术的要求 窄带信号的频谱如图2.21所示，对于下列三种情况，求并且画出在 $\pm F_s/2$ 范围内抽样信号的频谱。

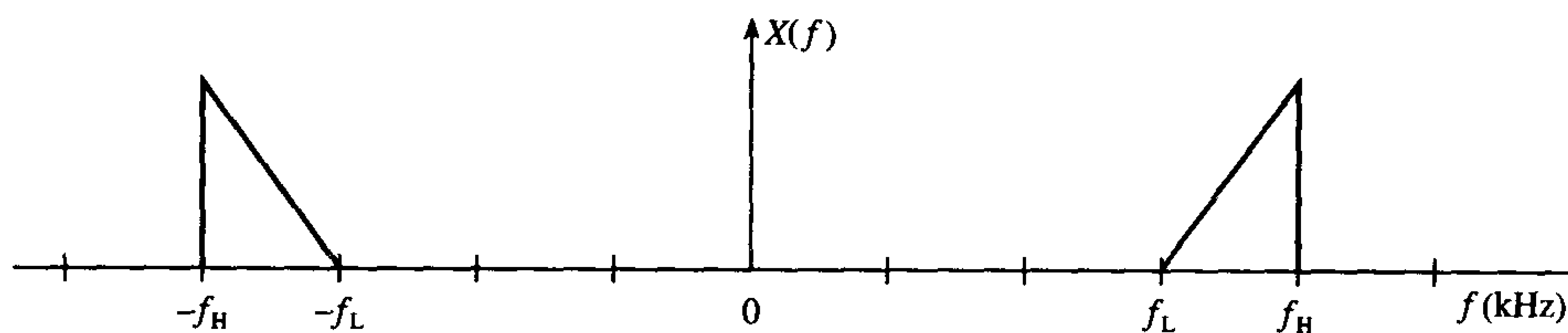


图2.21 窄带信号的频谱（例2.7）

(1) $\dfrac{f_H}{B} = 4$

(2) $\dfrac{f_H}{B} = 5$

(3) $\dfrac{f_H}{B} = 6.5$

假定信号的带宽是 $B = 4$ kHz，在每种情况信号以 $2B$ 的速率抽样。

解：

(1) 在这种情况下，信号的频谱如图 2.22(a)所示，以 $2B$ 抽样给出了 8 kHz 的抽样频率。抽样后信号的频谱可以通过将图2.22(a)信号的频谱与图2.22(b)抽样函数的频谱卷积而用图解的方式得到。

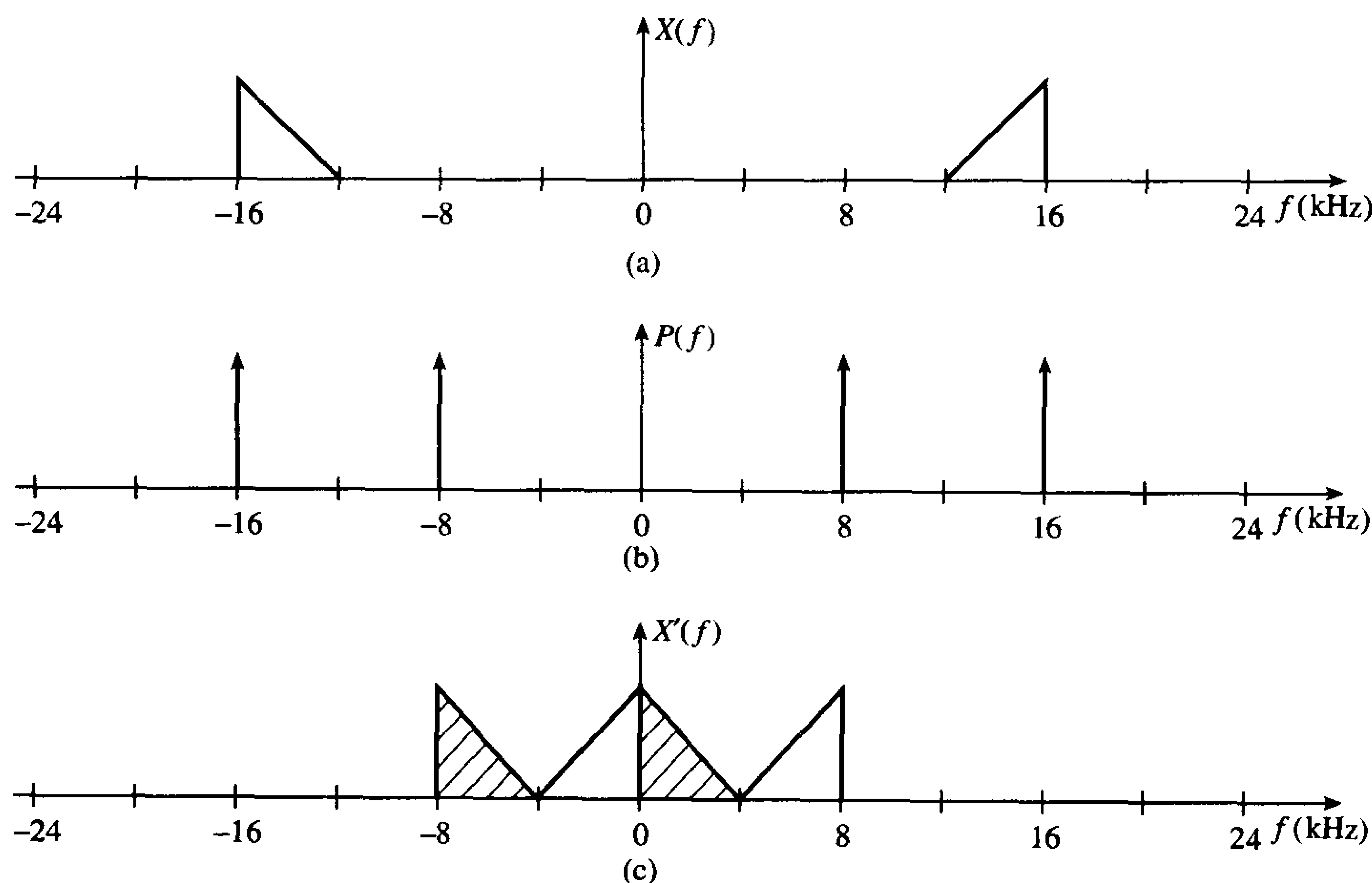


图 2.22　n 为偶数（$n=f_H/B=4$）时抽样后信号的频谱（例 2.7，情况 1）

我们固定信号的频谱，通过平移抽样函数的频谱来求卷积。在图解卷积过程中，我们通常在平移过程开始前首先将要平移的波形关于纵轴旋转。然而，抽样函数关于频率轴是对称的，旋转的结果将得到同样的波形，所以这一步是不必要的。

我们注意到，图 2.22(b)中在 –16 kHz 频率点刚好与信号频谱的负频部分对齐。因此，如果我们将抽样函数的频谱向右平移，–16 kHz的频率点将与信号频谱的负频部分卷积得到图 2.22(c)中的 0 ~ 4 kHz 的频谱，接着在图 2.22(b)中 8 kHz 的频谱点开始与信号频带 12~16 kHz 的频谱部分卷积，产生图 2.22(c)中 4 ~ 8 kHz 的频谱。

通过将抽样函数的频谱左移就可以得到抽样后信号频谱的镜像，由负频部分产生的抽样后信号频谱用影线表示。

注释： 比较原始信号 12 ~ 16 kHz 频谱，抽样后信号频谱在 0 ~ 4 kHz 的频谱部分被颠倒。除了这一部分以外，信号频带没有混叠。因此，通过合适的频谱混叠算法可以把信号恢复出来。

(2) 抽样频率还是 8 kHz，信号的频谱和抽样函数的频谱如图 2.23(a) ~ (b)所示。

和前面一样，固定信号的频谱，首先将抽样函数的频谱右移，然后再左移得到图 2.23(c)所示的抽样后信号的频谱。

同样，由负频率分量产生的抽样后信号的频谱用阴影线表示。

注释： 比较原始信号在 16 ~ 20 kHz 的频谱，抽样信号在 0 ~ 4 kHz 的频谱部分是信号在 16 ~ 20 kHz 的频谱直接平移过来的，信号频带没有混叠，可以恢复。

(3) 正如前一种那样，抽样频率为 8 kHz，信号的频谱和抽样函数的频谱如图 2.24(a) ~ (b)所示。和前面一样，我们将固定信号的频谱，首先将抽样函数的频谱右移，然后左移，得到图 2.24(c)所示的抽样后信号的频谱。

在图 2.24(b)中我们注意到在–24 kHz的频率点是在信号频谱负频部分的中间，在24 kHz的点是在 22 ~ 26 kHz 频谱部分的中间。因此，当我们将抽样函数的频谱向右平移时，在 –24 kHz 处的频率点将与信号负频部分卷积，在 24 kHz 将与信号正频部分卷积，得到

图2.24(c)的0～2 kHz的频谱。由负频分量得到的抽样后信号的频谱部分用虚线表示，而由正频分量得到的频谱用实线表示。

进一步平移4 kHz后，在图2.24(b)的16 kHz频谱点开始与信号频带正的部分卷积，产生中心在8 kHz处的频谱，如图2.24(c)的实线所示。而虚线是由–16 kHz频谱点产生的，通过将抽样函数的频谱向左平移，可以得到抽样后信号频谱的镜像。

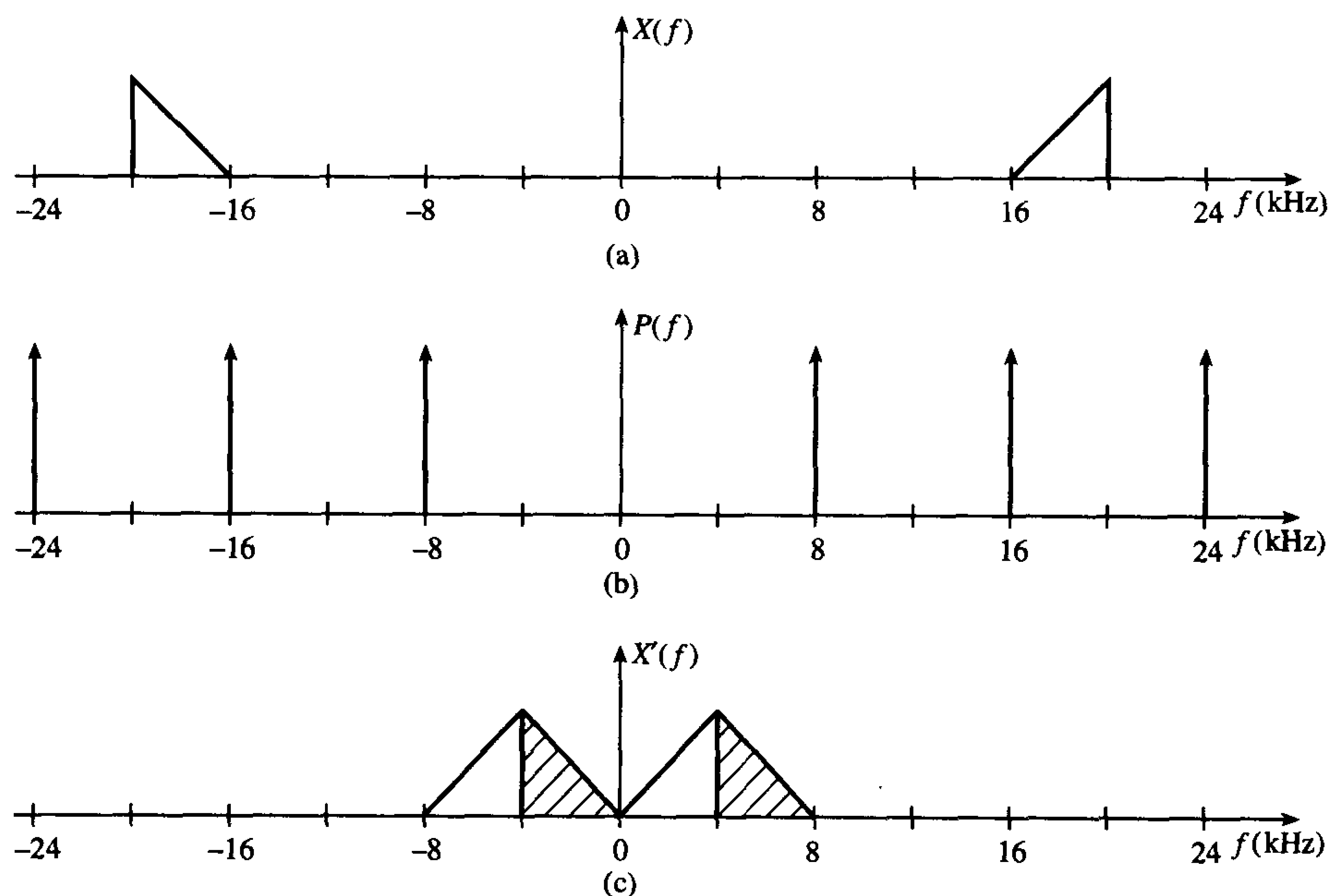


图2.23　n为奇数（$n = f_H/B = 5$）时抽样后信号的频谱（例2.7，情况2）

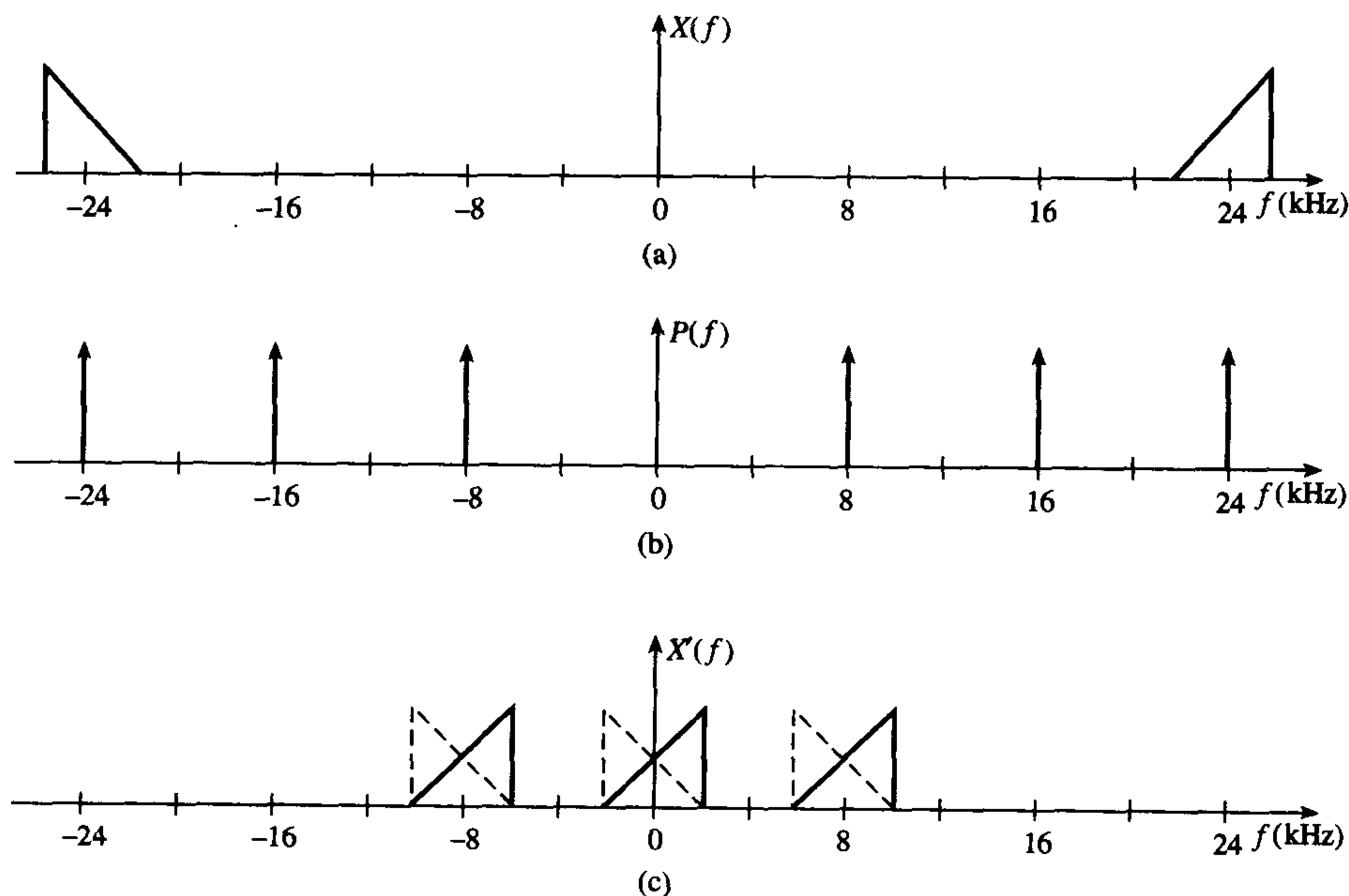


图2.24　抽样后信号的频谱（$n = f_H/B = 6.5$）（例2.7，情况3）

注释：由正频和负频产生的频谱的重叠指出存在混叠，在这种情况下，不能由 8 kHz 的抽样恢复信号分量。

2.3.2.3　扩展信号的带宽满足无混叠带通欠抽样的要求

我们已经看到，在整数带通抽样中，如果带沿频率是带宽的整数倍，那么就可以按照低得多的抽样频率（$2B$）对窄带 HF 信号进行抽样而且仍然避免混叠误差。

因此，在整数带通抽样中一个重要的参数是上沿频率f_H与带宽B之比（或者等效地为下沿频率f_L与带宽之比）：

$$n = \frac{f_H}{B} \tag{2.9a}$$

或者

$$n = \frac{f_L}{B} \tag{2.9b}$$

在两种情况下，我们都可以按照 $2B$ 的抽样率抽样而不会产生混叠，即

$$F_s = 2B \tag{2.10}$$

当比值是一个偶数时，抽样波形的频谱在基带域内是倒转的。

当 2.9a 式或 2.9b 式的 n 不是整数时，我们会发现存在混叠。通过扩展带沿频率或者中心频率使得 n 变成整数就可以避免混叠。例如，我们可以将下沿频率f_L扩展为f_1，使得

$$f_1 \leqslant f_L \tag{2.11a}$$

$$f_H = n(f_H - f_1) = nB' \tag{2.11b}$$

由 2.11b 式，我们可以写成

$$f_1 = \left(\frac{n-1}{n}\right) f_H \tag{2.12}$$

由 2.11a 式和 2.12 式，我们可以写成

$$\left(\frac{n-1}{n}\right) f_H \leqslant f_L$$

由此可以得到 n 的表达式：

$$n \leqslant \frac{f_H}{f_H - f_L} = \frac{f_H}{B} \tag{2.13}$$

因此，我们可以按 2.12 式扩展下沿频率来达到带沿频率和带宽之间所希望的关系，其中 2.12 式中的 n 是最靠近 2.13 式的整数。

可以证明，按照下式扩展上沿频率也可以达到目标：

$$f_2 = \left(\frac{n}{n-1}\right) f_L \tag{2.14}$$

其中 n 由 2.13 式给出，2.14 式的证明留给读者练习。

例 2.8　如何扩展带宽来避免混叠　在例 2.7（情况 3）中扩展下沿频率，确定避免混叠的最小抽样频率。

画出以新的抽样频率在抽样前后修正信号的频谱。

解：

上沿频率与带宽之比为

$$\frac{f_{\mathrm{H}}}{B}=\frac{26}{4}=6.5$$

如果我们令 $n=6$（最小最近的整数），那么可以将下沿频率化简为

$$f_{\mathrm{L}}'=\left(\frac{n-1}{n}\right)f_{\mathrm{H}}=21.66\ \mathrm{kHz}$$

新的带宽 B' 和抽样频率 F_{s}' 变成

$$B'=f_{\mathrm{H}}-f_{\mathrm{L}}'=4.34\ \mathrm{kHz}$$

$$F_{\mathrm{s}}'=2B'=8.68\ \mathrm{kHz}$$

抽样前后信号的频谱绘制留给读者进行练习，频谱图表明已经避免了混叠。

在某些应用中，通过改变中心频率而不是改变带宽，也是有可能达到无混叠整数带抽样的。数字无线电通信就是这样一种情况，其中本地IF（中频）是可以由设计者选择的。

2.4 均匀、非均匀量化和编码

抽样以后，可以采用均匀与非均匀量化和编码方法对模拟抽样值的幅度进行量化和编码，这取决于具体的应用。在生物医学和语音系统中常常采用均匀量化和编码，而在通信系统中由于需要压缩语音信号而广泛采用非均匀量化和编码（参看后面的章节）。

2.4.1 均匀量化和编码（线性脉冲编码调制（PCM））

在均匀量化和编码中，每一个模拟抽样值都赋予 2^B 值（参见图2.25）中的一个，其中 B 是ADC的位数。这个过程称为量化，由此引入的误差是无法消除的，误差大小是ADC位数的函数，近似等于LSB的一半（假定舍入）。例如，输入电压范围为 ±10 V 的12位ADC的LSB为 $20/2^{12}$ V，即 4.9 mV，那么量化误差为 2.45 mV。

对于 B 位的ADC，量化电平数是 2^B，电平之间的间隔，即量化步长 q 由下式给出：

$$q=V_{\mathrm{fs}}/(2^B-1)\approx V_{\mathrm{fs}}/2^B \tag{2.15}$$

其中 V_{fs} 为具有双极性输入信号的ADC的满刻度范围。对于上舍入或下舍入的情况，最大量化误差为 $\pm q/2$。对于幅度为 A 的正弦信号输入（信号的峰峰值刚好是ADC的输入范围），量化步长为

$$q=2A/2^B \tag{2.16}$$

每个抽样值的量化误差通常都假定为随机的，且在 $\pm q/2$ 的间隔上均匀分布，均值为零。在这种情况下，量化噪声功率即方差为

$$\begin{aligned}\sigma_e^2&=\int_{-q/2}^{q/2}e^2P(e)\,\mathrm{d}e\\&=\frac{1}{q}\int_{-q/2}^{q/2}e^2\,\mathrm{d}e=\frac{q^2}{12}\end{aligned} \tag{2.17}$$

对于正弦波输入，平均信号功率是 $A^2/2$。信号与量化噪声功率比（signal-to-quantization noise power ratio，SQNR）用分贝表示为

$$\text{SQNR} = 10\log\left(\frac{A^2/2}{q^2/12}\right) = 10\log\left(\frac{3\times 2^{2B}}{2}\right) = 6.02B + 1.76\ \text{dB} \tag{2.18}$$

这是理论上的最大值。在实际中，对于实际的输入信号，可达到的SQNR要小于这个值。然而，SQNR随位数B增加。实际的因素如速度、模拟信号固有的信噪比（SNR）以及成本都限制了位数的增加。例如，采用精度比待转换模拟信号的信噪比更好的转换器是没有必要的，因为这只是给出了噪声的精确表示。在许多 DSP 应用中，ADC 在 12 ~ 16 位之间是比较合适的。

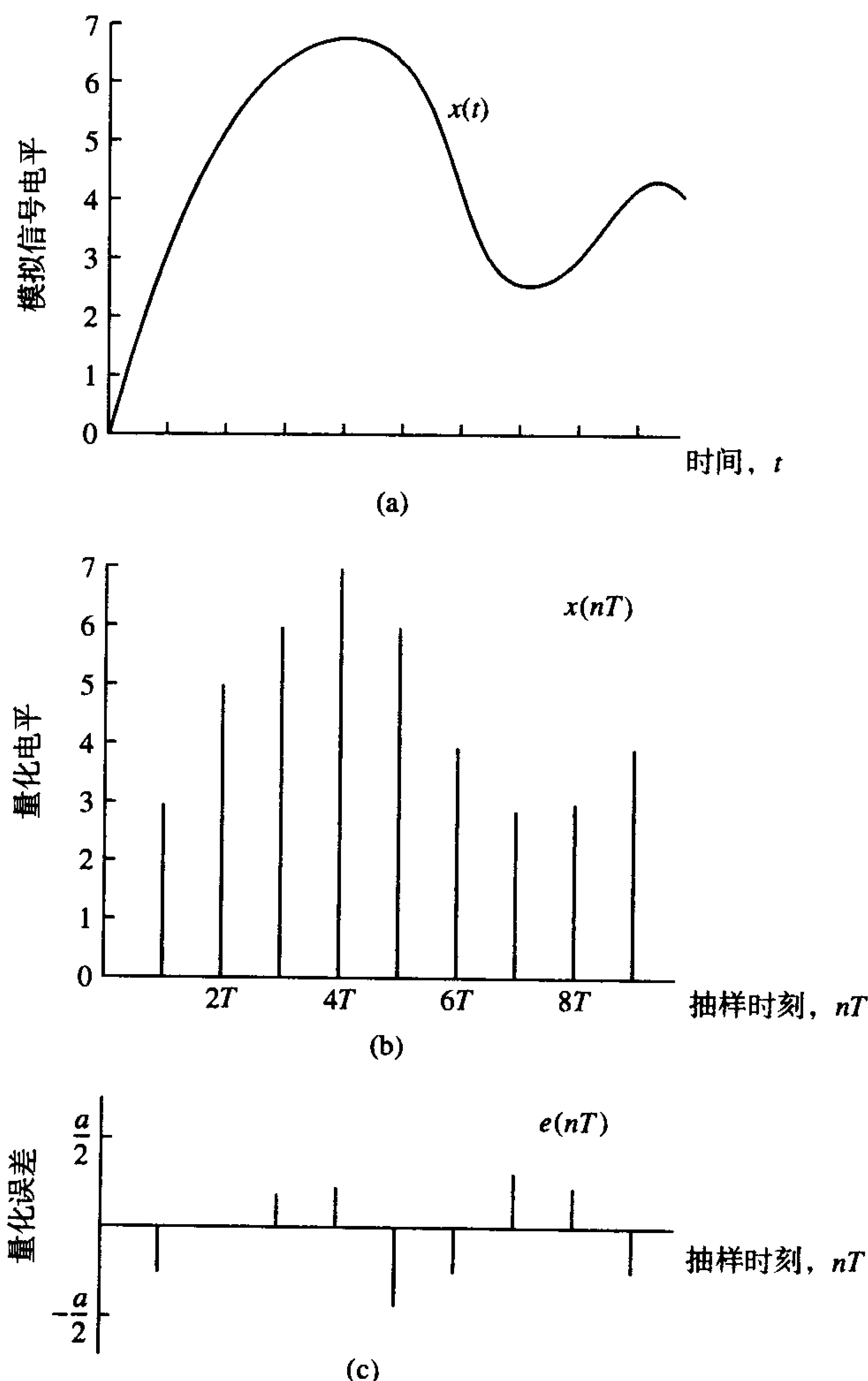


图 2.25　模拟信号抽样值的量化。(c)中的量化误差是由(a)中的信号样本值减去(b)中量化样本值（3 位量化器）得到的

在许多情况下，用二进制表示的数字抽样值接着将编码成适合进一步处理的形式。编码意味着给量化的抽样值赋予一个代码，在 DSP 中，最常用的形式是定点（补码）、浮点和块浮点（block floating point）表示。注意，同时进行抽样、量化和编码这三种操作是可能的，采用无抽样保持电路的 ADC 就是这样一种情况。

例 2.9 结合模数转换过程来解释动态范围、孔径时间的含义。

在例 2.2 中，如果 ADC 的动态范围大于 70 dB，抽样值数字化精度为$\frac{1}{2}$LSB，求

(1) ADC 的最小分辨率，用位数表示；

(2) 最大可允许的孔径时间，假定被数字化的信号的最高频率为 20 kHz。

解：

动态范围是模数转换器能够处理的最大信号电平与最小信号电平之比，动态范围常常根据转换器的位数用分贝表示为

$$D = 20\log_{10}2^B \tag{2.19}$$

在某些应用中，动态范围是根据信号功率定义的。例如，在数字音频系统中，动态范围定义为能够从噪声功率中识别出的最大信号功率与最小信号功率之比。

当 ADC 单独使用时，孔径时间基本上就是 ADC 的转换时间，也就是模拟输入信号必须维持稳定以便保证精确的转换的持续时间。就抽样保持而言，它是在保持命令后达到保持所要求的时间。

(1) 应用 D 的表达式，由 $B = 11.62$，我们得

$$70 = 20\log_{10}2^B$$

取 $B = 12$（最近的整数）。

(2) 最大可允许的孔径时间 τ 为

$$\tau = 1/2^{B+1}\pi f_{\max} = 1/(2^{13}\times\pi\times 20\times 10^3)\text{ s} = 1.94\text{ ns}$$

这样小的孔径时间要求在在 ADC 之前采用抽样保持电路。

2.4.2 非均匀量化和编码（非线性 PCM）

到目前为止，我们讨论的线性模数转换过程有时也称为线性 PCM。在这样的转换器中，量化噪声电平直接与 ADC 的位数有关，这样的转换器非常适合于信号的幅度是均匀的应用场合。在这样的应用中，A/D 转换器的字长大小不是主要关心的问题。而在信号幅度不是均匀分布的应用场合（如电话），为了精确地表示数据要求很多位数，因而这不是有效的方法。

有些信号（如语音）包含了小的和大的幅度，而小幅度更多一些。这样，均匀量化对语音是不合适的，非均匀量化对低电平信号能够在位数相同的情况下比均匀量化提供了更多的量化电平。在电话中这意味着轻声说话的人和大声说话的人都是能够适应的。

在电话中（公用和私用电话网络），标准的非均匀量化是由语音的幅度分布知识来确定的。语音波形以 8 kHz 的速率抽样，给定一个 64 kb/s 的系统，对每一个抽样值量化并用 8 位进行编码。语音抽样值的幅度在传输前被对数压缩成 8 位，在接收端再将压缩的数据扩展。

压缩和扩展语音信号的过程称为压扩（companding，取自 COMpressing 和 exPANDING 的缩写），其过程在图 2.26 中给出。在实际中，压扩是由多媒体数字信号编码器或者组合多媒体数字信号编码器（combo-codec，PCM 多媒体数字信号编码器及抗混叠和抗镜像滤波器组合而成）执行的，多媒体数字信号编码器适合数字电话的每一个语音信道。

在带有多媒体数字信号编码器的现代数字电话中，在链路中为了允许数据用 DSP 处理，逆压扩处理是必需的。压缩的 PCM 被转换成线性 PCM 数据，经过 DSP 处理后，数据又转回到非均匀 PCM。逆压扩运算由 DSP 芯片用查表的方法（如德州仪器的 TMS320 系列和摩托罗拉的 56000 系列）来执行，或者实时执行压缩和扩展算法来实现。

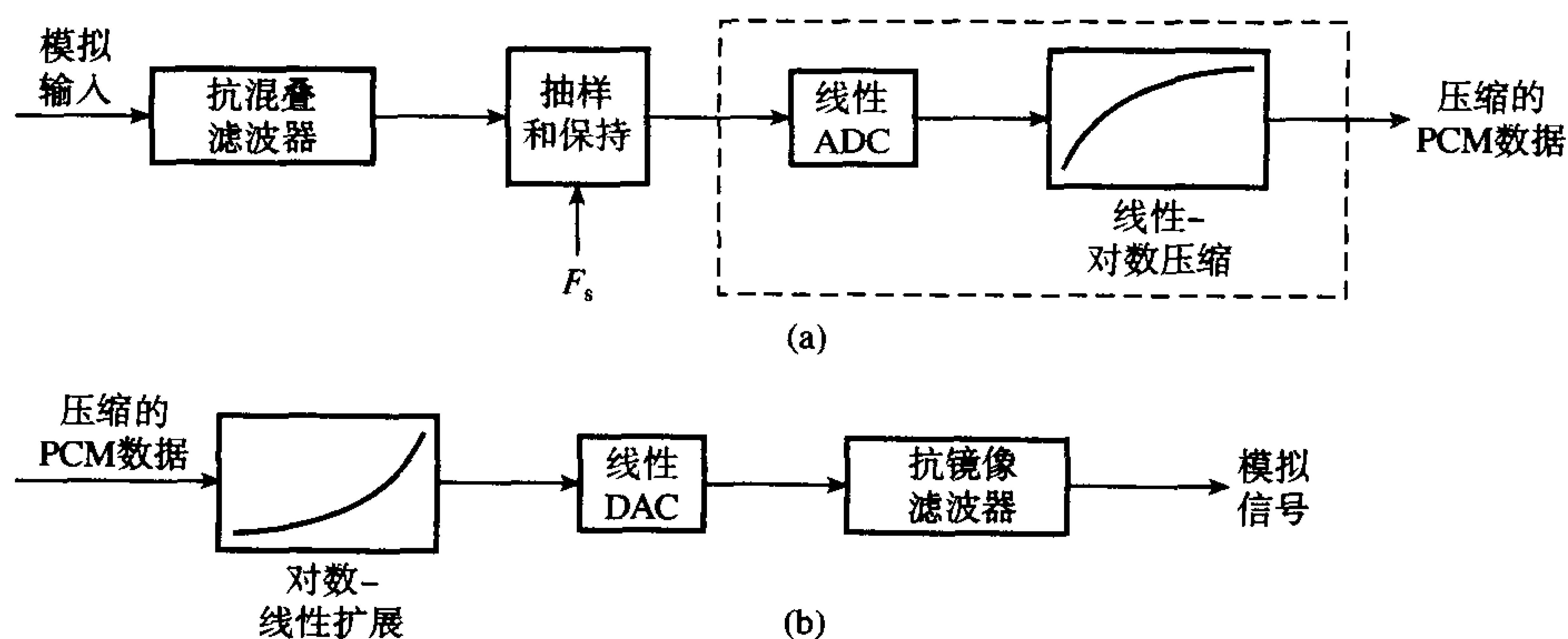


图 2.26　非均匀模数和数模转换

2.4.2.1　压扩方法：μ 律和 A 律 PCM

在电话中为了达到非均匀量化采用了两种国际标准：μ 律标准（在美国和日本采用）和 A 律标准（主要在欧洲采用）。两种标准都把语音压缩成 8 位，与线性 ADC 的 14 位等效。

对于 μ 律，图 2.27 给出了压扩特性和方程的定义，用八位带符号的幅度字来表示每个抽样值。压扩特性近似用八条直线线段表示（参见图 2.27）。从图中可以看出，大的输入信号抽样值被压缩成均匀输出值，输入信号的步长在段之间被接连翻倍，这使得它易于在线性和非均匀 PCM 之间进行转换。μ 律 PCM 由八位字组成，MSB 是符号位，下面三位表示段号，最后四位表示段内的位置。在实际中，在传输前将位颠倒，以便增加 1 的密度，从而有助于时钟的恢复和误差校正。这是必需的，因为语音是低能量的信号。非均匀量化的信号与量化噪声之比与 14 位线性 ADC 是相当的。

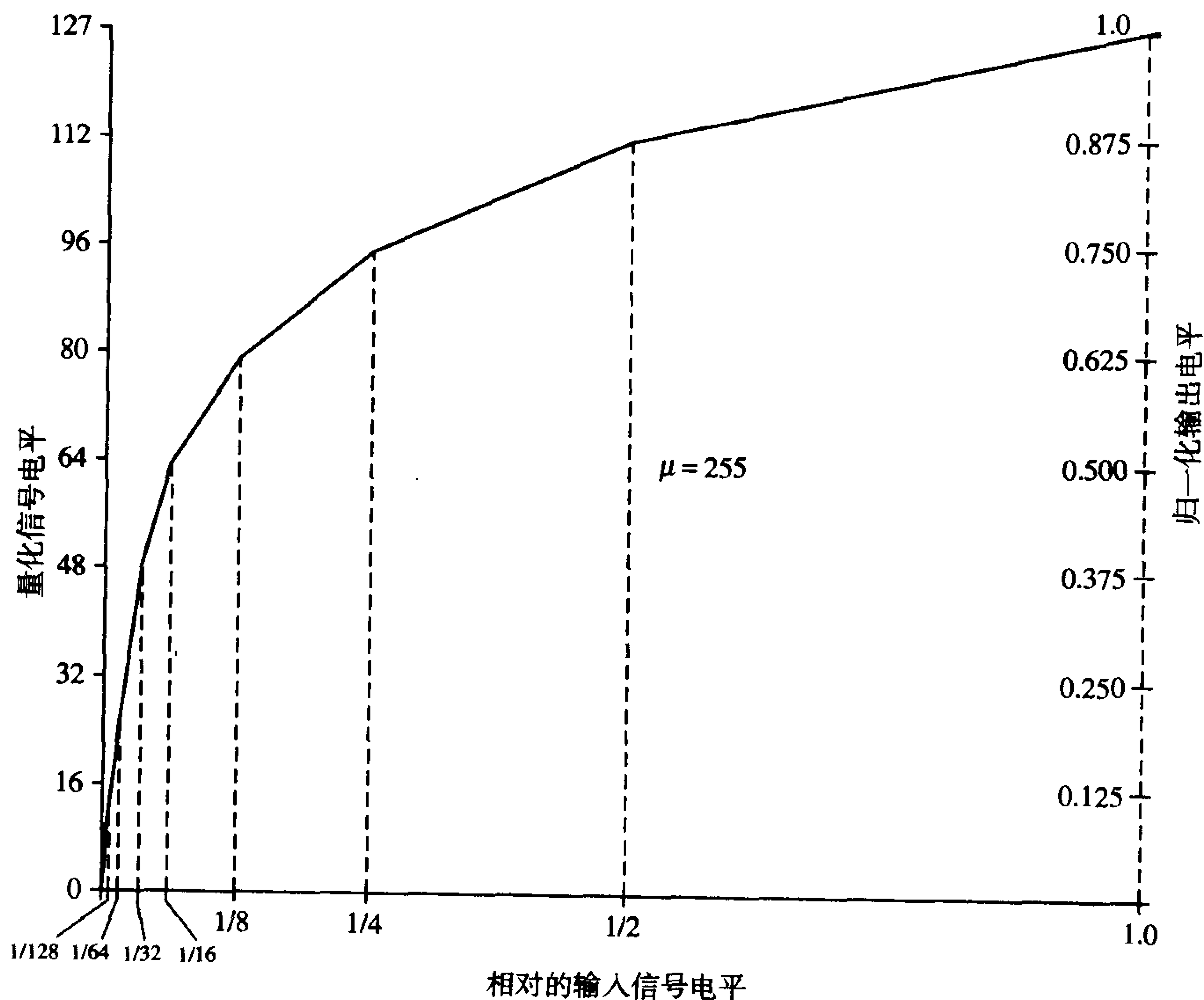


图 2.27　μ-255 律的压扩特性（ITU, 1998），特性由下面的方程定义：$F(x)=\mathrm{sgn}(x)\dfrac{\ln(1+\mu|x|)}{\ln(1+\mu)}$，其中 $\mu=255$，x 是归一化输入信号，sgn 是符号函数，$F(x)$ 是压缩后的输出信号

图2.28给出了A律特性，它类似于μ律特性。然而，A律对小信号的保真度较差，但有较好的动态范围特性。

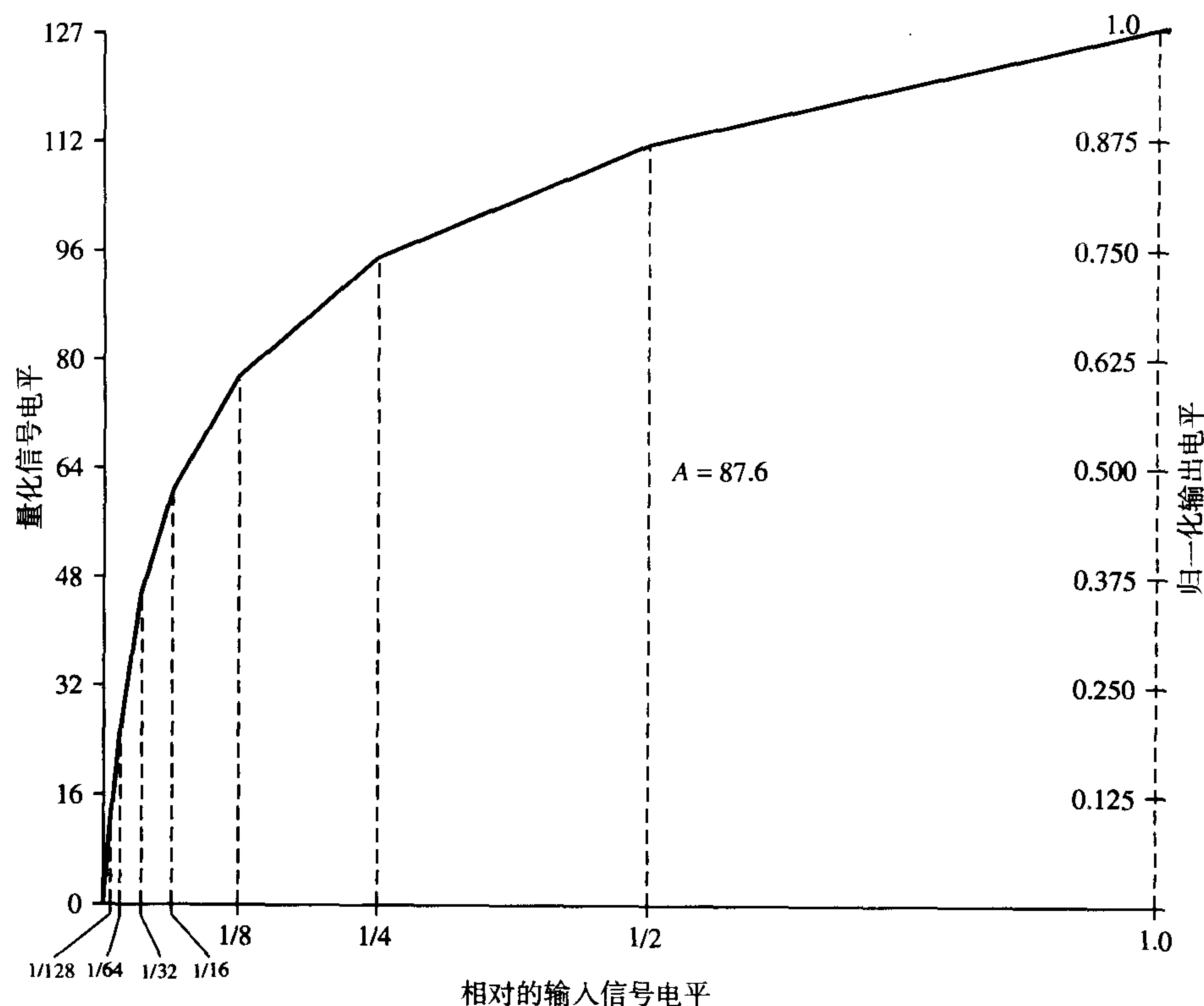


图2.28 A-255律压扩特性（ITU, 1998），特性由下面的方程定义：

$$F(x)=\mathrm{sgn}(x)\frac{A|x|}{1+\ln(A)},\ 0\leqslant|x|<1/A;\ F(x)=\mathrm{sgn}(x)\frac{1+\ln(A|x|)}{1+\ln(A)},\ 1/A\leqslant|x|<1$$

其中$A=87.6$，x是归一化输入信号，sgn是符号函数，$F(x)$是压缩后的输出信号

2.4.2.2 自适应差分脉冲编码调制（ADPCM）

非均匀PCM将语音信号抽样值的每一个数据样本量化成8位，这种方法没有利用语音信号的冗余特点。在ADPCM（CCITT推荐G.726, 1990; CCITT, 1989）中，每个样本的值得到调整，这依赖于具体的样本值。这样，将表示每个样本的位数由8位减少到4位（8 kHz × 4位= 32 kb/s），ADPCM发射预测样本值与实际样本值之间的差。在实际中，ADPCM代码转换器可能插入到PCM系统中，以便增加它的声音信道容量。ADPCM编码器接收PCM值作为输入，ADPCM解码器输出PCM值。

对于语音编码存在许多标准，这些标准在通信工业的许多服务和应用中起到了基准的作用。在大多数情况下，强调的是数据率的减少。例如，GSM移动电话系统的语音速率为13.2 kb/s，CCITT G721 ADPCM标准则以32 kb/s的速率工作。

2.5 A/D转换中的过抽样

2.5.1 引言

实际上，过抽样意味着对输入信号以比奈奎斯特频率高得多的频率进行抽样。实际的抽样频率和奈奎斯特频率之比称为过抽样比（假定低通信号）：

$$过抽样比 = F_s/2f_{max} \tag{2.20}$$

在现代DSP的许多领域中的趋势是采用过抽样，以便利用抽样定理的实际含义。在A/D接口时，过抽样的主要好处是：(1) 简化了抗混叠滤波器；(2) 支持可变截止频率的抗混叠滤波器（每一个截止频率要求一个不同的抽样频率）；(3) 通过将量化噪声分布在较宽的频带来降低噪声水平，这使得使用较少位数的 ADC 达到与高分辨率 ADC 相同的 SNR 性能。

2.5.2　过抽样和抗混叠滤波

在高保真数字系统中，保持较低的混叠误差电平的需要常常被迫采用相对复杂的模拟抗混叠滤波器。在多通道系统中，每一个模拟通道都必须采用不同的抗混叠滤波器。因为这样的滤波器是不能多路切换的，在模拟通道数较多的地方就显得很昂贵（例如，在生物医学中可能要求64个通道）。此外，抗混叠滤波器的相位匹配（在所有通道上保留信号分量之间的关系）可能很难达到。

过抽样技术允许我们克服许多这样的问题，考察一下抽样后信号的频谱就可以看出这是显而易见的。抽样频率越高，像频分量和基带分量就分得越开（例如，参见图2.29）。在第9章中，我们将看到过抽样技术如何与多速率DSP组合以满足高保真系统的要求。

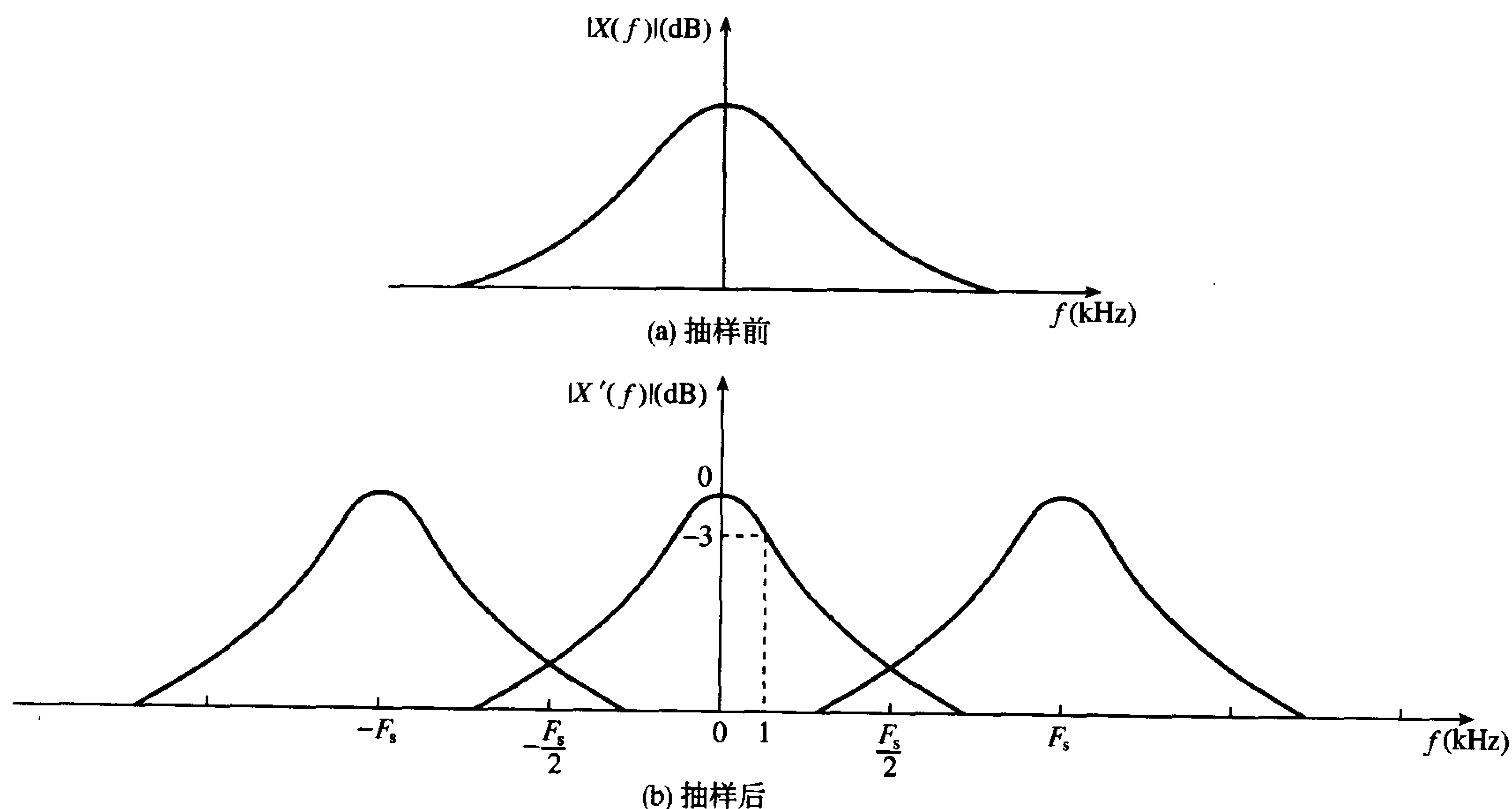


图 2.29　输入信号的频谱

例 2.10

(a) 用于收集神经生理学数据的通用多通道数据采集系统的技术要求：每个模拟通道由用户分别构造，带沿频率在 0.5 Hz 到 500 Hz 之间，可选的抽样频率在 1 Hz 到 5 Hz 之间，在通带内，最大允许的波纹是 0.5 dB，像频分量低于信号分量 40 dB。

为了满足以上要求，解释你将会采取的策略，你的答案应该阐述如下几点：

(i) 对特定应用问题的考虑；

(ii) 为了满足有效和经济（根据成本/器件价格）的要求，在该应用中如何使用过抽样技术。

(b) 假定在(a)的系统中所有通道采用相同的抗混叠滤波器，每个抗混叠滤波器具有下列巴特沃斯特性：

$$A(f) = \frac{1}{\sqrt{1+\left(\frac{f}{f_c}\right)^6}}$$

其中$f_c = 3$ dB 为滤波器的截止频率。

(c) 借助抽样前后数据频谱图，求

(i) 截止频率f_c；

(ii) 合适的常规抽样频率 F_s。

解释你的结果。

解：

(a) 高分辨率ADC/DAC速度低，限制了最大可达到的抽样频率，这是许多实时应用的主要瓶颈。为了克服这一点，可能需要采用多ADC/DAC器件或者多速率DSP技术。

在输出端逐步降低信号高频分量的sin *x*/*x*效应，可以通过采用具有*x*/sin *x*响应的数字滤波器进行补偿，其他合理的答案是可接受的。

(b) (i) 为了保存信号中临床感兴趣的信息，应该尽可能地使幅度和相位失真小。应该保存各通道特征之间的时间关系，采用相同的具有良好的幅度/相位响应的抗混叠滤波器是可期待的。

(ii) 为了减少器件价格 / 成本和系统的PCB尺寸，所有64个通道应该采用相同的、简单的抗混叠滤波器。那么，通道应该以常规的固定速率进行过抽样。采用多速率技术，高的相同的抽样率可以减低到期望的速率。至少应该采用二阶巴特沃斯滤波器来避免过高的常规抽样率。

(c) 从技术要求考虑和抽样前后数据的频谱，我们求得

(i) 为了满足技术要求，在0和500 Hz的幅度误差应该满足下列准则：

$$20\log\left[1+\left(\frac{500}{f_c}\right)^6\right]^{\frac{1}{2}} \leqslant 0.5\,\mathrm{dB}$$

其中，我们假定了三阶巴特沃斯滤波器，截止频率为f_c。

解f_c，我们求得

$$f_c = \frac{500^6}{0.122} \approx 710\,\mathrm{Hz}$$

为了允许在后级附加误差以及为了方便起见，令f_c= 1000 Hz（这等价于0.26 dB的最大波纹）。

(ii) 带限每个通道后，抽样后数据的频谱具有图2.29所示的形式。

现在选择F_s，使500 Hz处的混叠误差电平至少下降40 dB，即

$$20\log\left[1+\left(\frac{F_s-500}{1000}\right)^6\right]^{\frac{1}{2}} \geqslant 40\,\mathrm{dB}$$

求解F_s得F_s= 5141.5 Hz。

为了允许有效降低抽样率，要求小心选择常规抽样频率F_s。一个可能的选择是8192 Hz，它允许将常规抽样率通过一个简单的整数因子而降低。

2.5.3 过抽样和ADC分辨率

对输入信号进行过抽样将量化噪声能量散布在很宽的频带范围内，因此降低了感兴趣的频带内的噪声电平，提高了ADC的分辨率。在数字音频中利用这一点来达到所谓的一位（single-bit）或者过抽样ADC。首先让我们复习一下量化和量化误差的基本概念。

2.5.3.1　量化和量化误差

在传统的A/D过程中，每个信号抽样值被量化成2^B电平中的一个，并且用B个二进制位表示，其中B是ADC的位数，量化引入了误差，它是ADC位数的函数。

量化噪声（均匀分布，均值为零）功率为

$$\sigma_e^2 = \frac{q^2}{12}$$

其中q是量化步长。线性ADC在理论上的最大信号量化噪声比（SQRN）为

$$\mathrm{SQNR} = 6.02B + 4.77 - 20\log(A/\sigma_x)\ \mathrm{dB} \tag{2.21}$$

其中$\pm A$是ADC的输入范围，σ_x是输入信号的均方根值。对于峰值幅度A刚好充满ADC范围$\sigma_x = \frac{A}{\sqrt{2}}$的正弦波输入，$20\log\left(\frac{A}{\sigma_x}\right) = 3.01\ \mathrm{dB}$，所以2.21式化为熟悉的形式：

$$\mathrm{SQNR} = 6.02B + 1.7\ \mathrm{dB} \tag{2.22}$$

例如，对于双极性的线性16位ADC，输入范围为±5 V，量化步长为$q = \frac{10\ \mathrm{V}}{2^{16}-1} = 0.152\ \mathrm{mV}$，最大量化误差为$\frac{q}{2} = 76\ \mu\mathrm{V}$，SQNR = 98 dB。

练习

峰峰值为10 V的正弦信号用12位ADC数字化，假定线性量化，求

(1) 量化步长；

(2) 量化噪声功率；

(3) 理论的最大信号量化噪声比（maximum signal-to-quantization noise ratio）。

2.5.3.2　过抽样和量化噪声功率

由A/D转换过程引入的固有量化噪声功率由下式给出：

$$\sigma_e^2 = \frac{q^2}{12} = \frac{2^{-2(B-1)}}{12}\ （归一化） \tag{2.23}$$

其中B是ADC字长（包括一位）。

对于足够大的或随机的模拟输入信号，量化噪声能量均匀分布在有效频带内，即0到$F_s/2$，其中F_s是抽样频率。在这种情况下，量化噪声的功率谱密度$P_e(f)$为（参见图2.30）

$$P_e(f) = \frac{\sigma_e^2}{F_s} \tag{2.24}$$

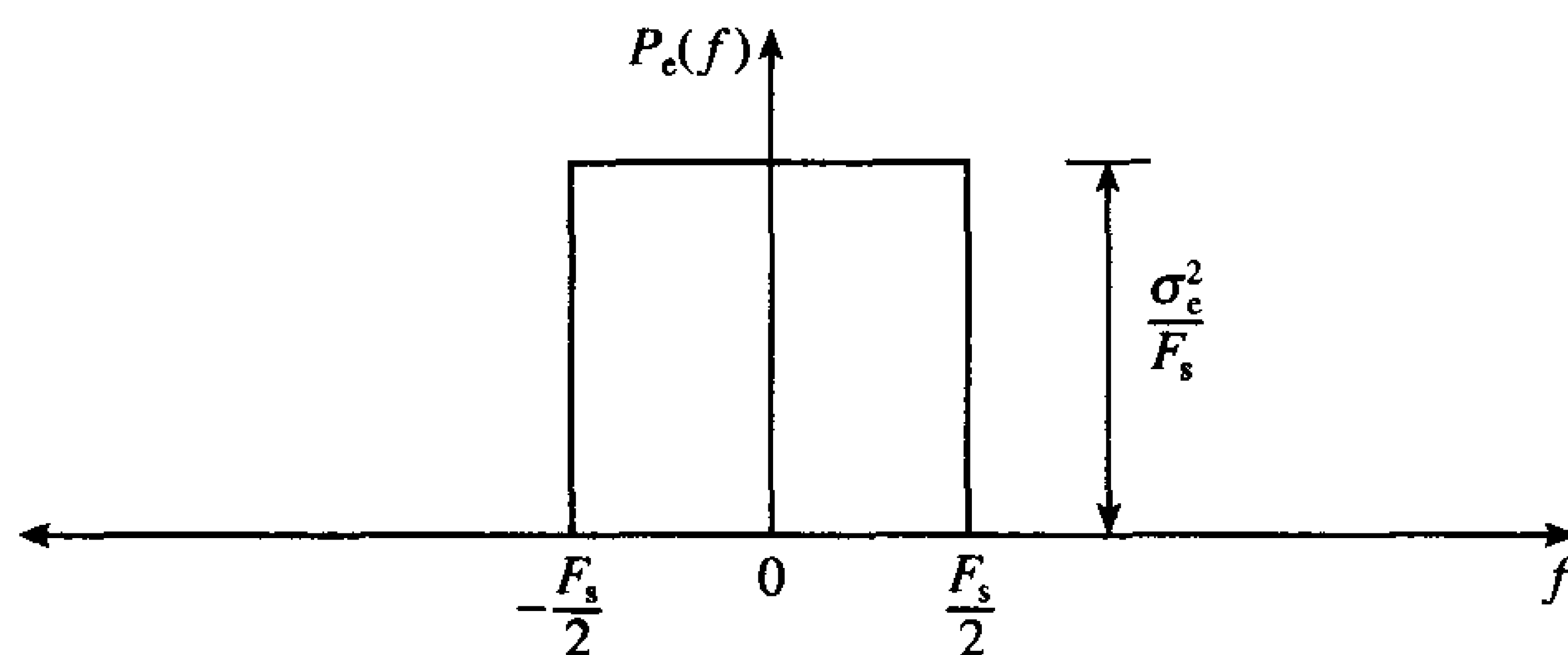
因此，通过以高的抽样率对输入数据抽样，使量化噪声能量分布在较宽的频带内，可以增加ADC的有效分辨率，这样降低了感兴趣频带内的噪声电平。这就是过抽样的含义。

参考2.20式，对于奈奎斯特率转换器，$f_{max} = F_s/2$，所以整个带内噪声功率由图2.30(a)的面积给出，即σ_e^2。对于过抽样转换器（参见图2.30(b)），部分量化噪声功率落在希望的频带之外（由于$f_{max} < F_s/2$），带内噪声小于奈奎斯特率转换器的带内噪声。

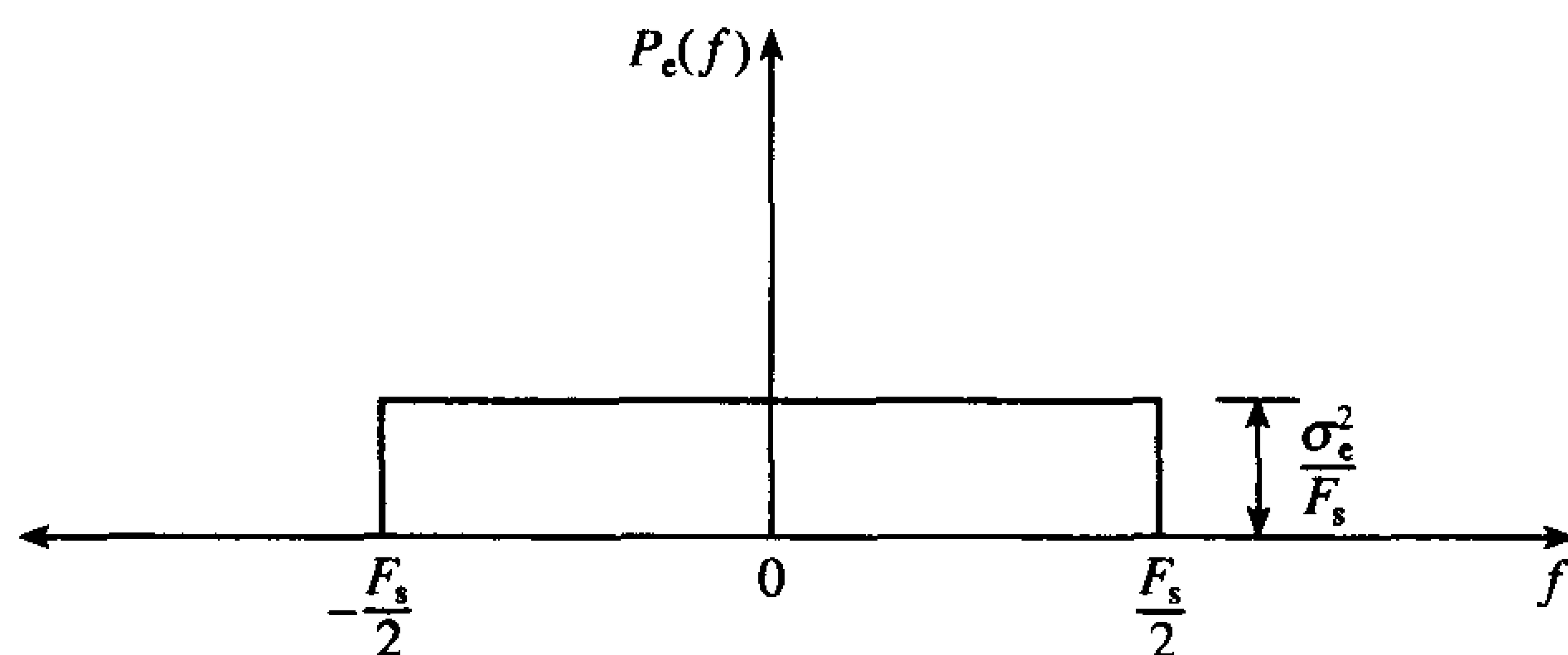
过抽样转换器的带内噪声功率由下式给出：

$$P_e = \int_{-f_{max}}^{f_{max}} P_e(f)\,df = \frac{2f_{max}}{F_s}\sigma_e^2 \tag{2.25}$$

因此，当我们对带限信号进行过抽样的时候，在信号频带内的量化噪声能量随过抽样比而降低。在实际中，为了易于实现，选择过抽样比为2的整数幂。



(a) 奈奎斯特率转换器



(b) 过抽样转换器

图2.30 量化噪声功率谱密度（两种转换器总的噪声功率是相同的，但是过抽样转换器的噪声功率散布在宽得多的频率范围内，使带内噪声功率电平降低了）

例2.11

(a) 一个音频系统处理信号，信号的基带为0到20 kHz。求过抽样比和用12位转换器能够达到16位转换器具有的性能所需要的最小抽样频率。

(b) 数字音频系统采用过抽样技术，并且用8位双极性奈奎斯特率转换器对模拟信号进行数字化，模拟信号的频率范围为0～4 kHz。如果抽样率为40 MHz，估计转换器的有效分辨率（用位数表示）。解释与此方法有关的实际问题。

解：

(a) 在奈奎斯特频率处（即$F_s=2f_{max}$），12位和16位转换器的归一化带内量化噪声功率分别为

$$\sigma_1^2 = 2^{\frac{-2(B_1-1)}{12}} \quad （其中 B_1=12）$$

$$\sigma_2^2 = 2^{\frac{-2(B_2-1)}{12}} \quad （其中 B_2=16）$$

为了用12位的ADC达到16位的性能，我们需要对输入到12位转换器的信号进行过抽样，以减少带内量化噪声功率。带内量化噪声功率随过抽样因子而减少，

$$\sigma_1'^2 = \frac{2f_{max}}{F_s}\sigma_1^2$$

使新的带内噪声功率等于16位ADC的带内噪声功率，我们有

$$\frac{2f_{max}}{F_s}\sigma_1^2 = \sigma_2^2$$

因此，

$$\frac{2f_{\max}}{F_s}=\frac{\sigma_2^2}{\sigma_1^2}=\frac{2^{-2(B_2-1)}}{2^{-2(B_1-1)}}=2^{-2(B_2-B_1)}=\frac{1}{256}$$

于是，过抽样比为

$$F_s/(2f_{\max})=256,\ \text{即}\ F_s=10.24\ \text{MHz}$$

(b) 带内量化噪声随过抽样比减少，过抽样比为

$$\frac{40\,000}{2\times 4}=5000$$

由$\frac{\sigma_1^2}{\sigma_2^2}=\frac{2^{-2(B_1-1)}}{2^{-2(B_2-1)}}=2^{2(B_2-B_1)}=5000$以及$B_1=8$位，我们求得ADC的分辨率$B_2$大约为14位。

在前面的例子中为了用低分辨率的ADC达到期望的分辨率，仅仅依靠过抽样技术自身并不是很经济的，因为这样的分辨率常常要求很高的抽样频率。当前的技术可能并不支持这么高的抽样频率。

在实际中，过抽样与噪声整形（noise shaping）滤波器组合在一起，噪声整形滤波器将量化噪声信号频带之外的高频，在高频将其滤除。过抽样ADC的原理在下一节进行介绍。

2.5.4　过抽样的应用——一位（过抽样）ADC

在高保真DSP系统的技术要求中，如数字音频系统要求高品质、高分辨率和高速（且便宜）ADC，这些要求用常规的逐次逼近或双斜率（dual slope）转换器难以满足，因为这些常规的转换器的误差与这些转换器的模拟部分有关（如ADC的补偿网络、抗混叠滤波器和抽样保持电路）。

一位或者更为贴切的是过抽样ADC并不要求抽样保持放大器，并且使用简单的甚至无抗混叠滤波器。所以，常规转换器中大多数的误差在过抽样ADC中都没有。

有两项技术使一位ADC成为可能：

- 过抽样，将量化噪声能量散布在很宽的频带内，因而降低了感兴趣的频带内的噪声电平；
- 噪声整形，将大部分噪声移到信号频带之外的高频，在高频用数字滤波方法将其滤除。

图2.31描述了过抽样ADC的概念，模拟输入信号被过抽样（如64倍），将量化噪声功率散布在一个较宽的频带内。然后，抽样的数据进行噪声整形，将噪声功率移到远离信号的高频端，再经过抽取（decimation）将抽样频率降至奈奎斯特频率。这个过程也起到从单比特流变换到多比特数据流的作用，抽取将在第9章详细介绍。

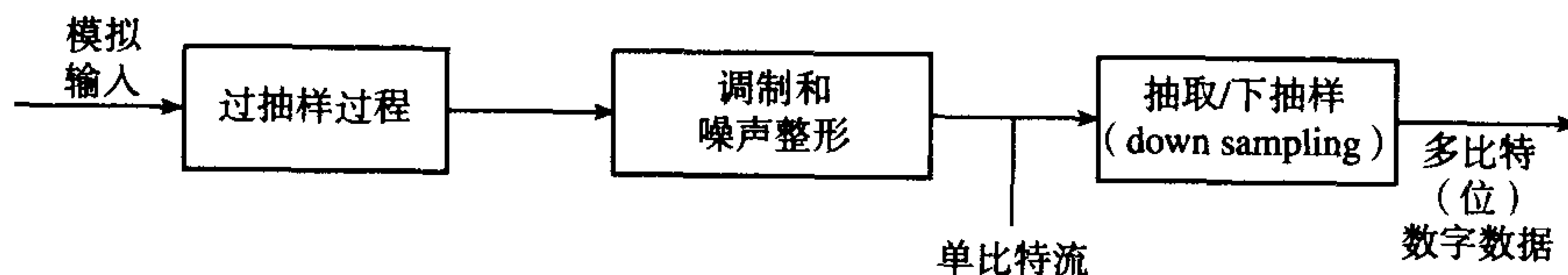


图2.31　过抽样（一位）A/D转换的原理

达到噪声整形的最有效的方法之一是Σ-Δ（sigma delta）调制。图2.32给出了一阶Σ-Δ调制器（SDM），它由积分器、一位量化器（即用比较器实现的1比特ADC）以及在反馈支路安排的一位DAC组成。模拟输入信号$x(t)$以很高的速率进行抽样，然后被量化成含有很高量化噪声的单比特流。通过选择积分器的合适特性，对噪声频谱进行整形，使大部分噪声能量上移，并移到信号频带之外。

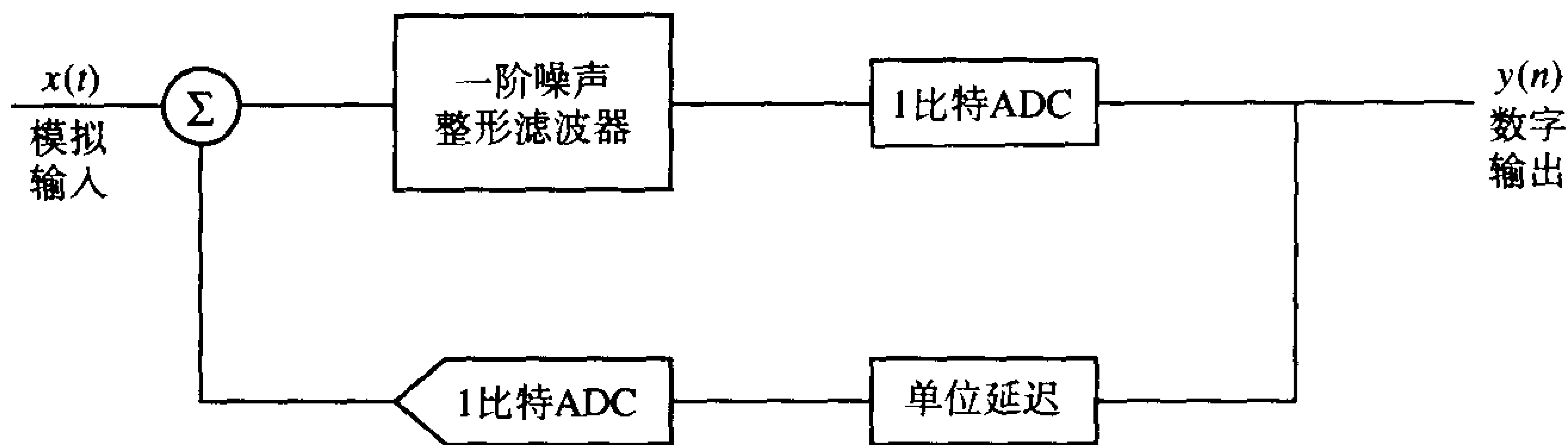


图 2.32 一阶 Σ-Δ 调制器

为了理解调制器是如何对量化噪声整形的，我们考虑图 2.33 所示的一阶 Σ-Δ 调制器的 z 域模型，其中我们假定噪声抽样值是不相关的。

根据图 2.33，输出的 z 变换为

$$\begin{aligned} Y(z) &= E(z) + [X(z) - Y(z)z^{-1}]\left(\frac{1}{1-z^{-1}}\right) \\ &= X(z) + E(z)(1-z^{-1}) \\ &= X(z) + E(z)H_n(z) \end{aligned} \tag{2.26}$$

其中

$X(z)$ ＝输入信号的 z 变换
$Y(z)$ ＝比特流输出的 z 变换
$E(z)$ ＝量化噪声的 z 变换
$H_n(z) = (1-z^{-1})$是噪声的传递函数

2.26式清楚地表明了输出变换等于输入变换加上经噪声传递函数修正后的量化噪声变换组成。量化噪声传递函数（$1-z^{-1}$）实际上是在dc（直流）有一个零点的高通滤波器，它的作用就是将量化噪声能量移到高频端，参见图 2.34。

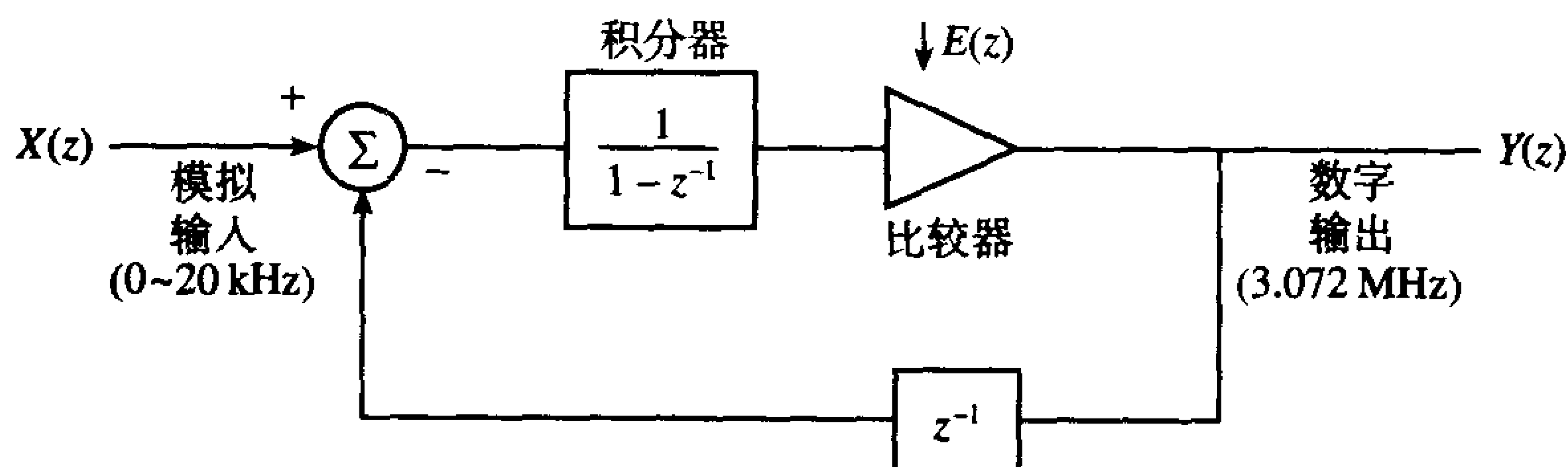


图 2.33 一阶 SDM 的 z 域模型

对于一个输入带限为$f_{\max}$的系统，噪声整形后的带内噪声功率为

$$\sigma_n^2 = \int_{-f_{\max}/F_s}^{f_{\max}/F_s} |H_n(f)|^2 P_{\mathrm{d}}\,\mathrm{d}f \tag{2.27}$$

很显然，SDM 的性能取决于过抽样比以及 SDM 对噪声谱整形的能力。对于一阶 SDM，抽样率翻倍，信噪比 SNR 将增加 9 dB，其中有 6 dB 是由噪声整形贡献的，3 dB 是由过抽样贡献的。增加噪声传递函数（即积分器）的阶数可以进一步减少量化噪声。可以证明，对于N阶 SDM，输出变换函数为

$$Y(z) = X(z) + E(z)(1-z^{-1})^N \tag{2.28}$$

它提供了一种 6N dB/倍频程噪声滤波器。遗憾的是，当 N >3 时，由于大的相移，调制器的稳定性得不到保证。对于阶数大于2的SDM，需要采用一种特殊的结构来避免不稳定。我们称其为MASH结构，图 2.35 给出了三阶 SDM 的 MASH 结构。

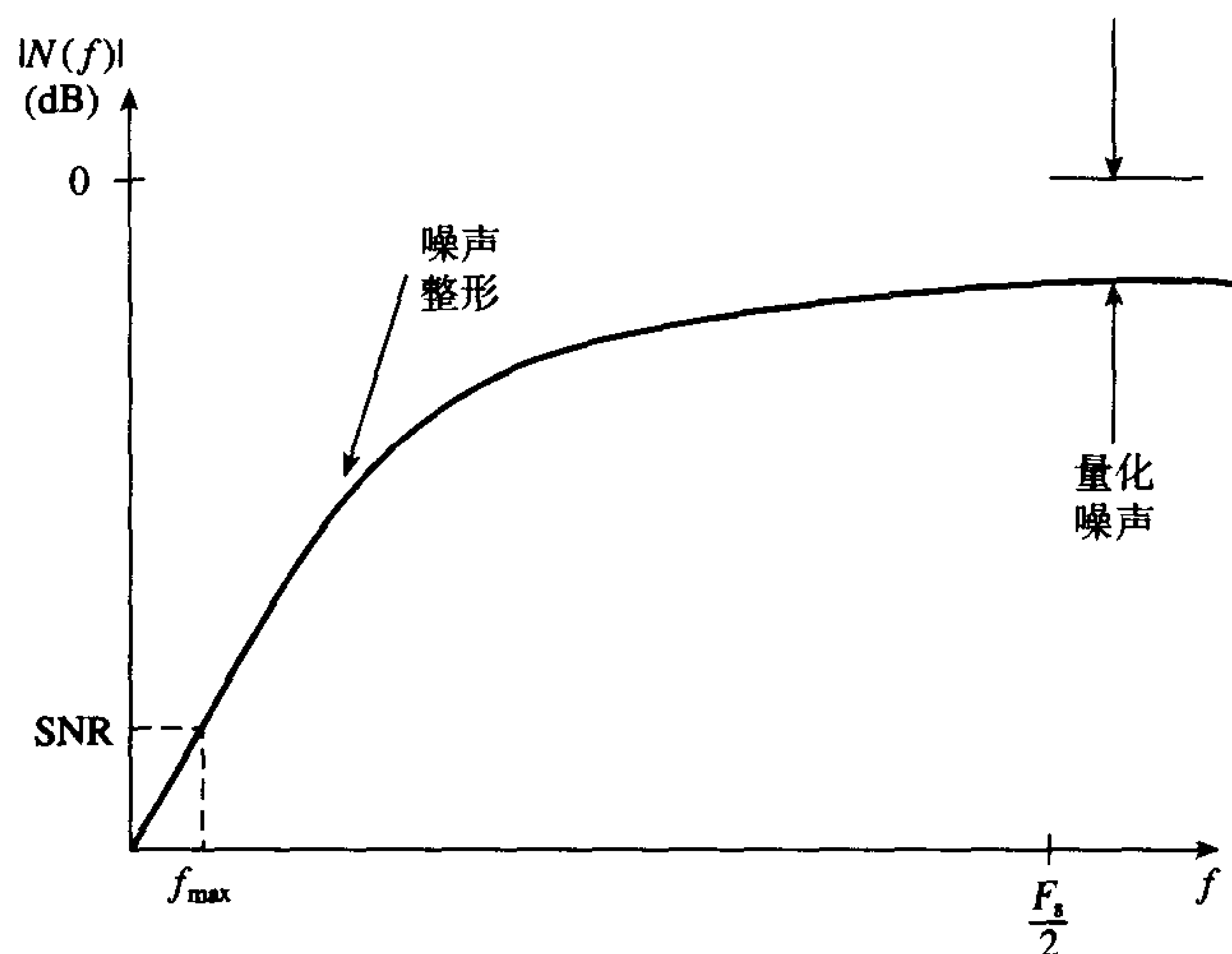


图 2.34　噪声整形对量化噪声的影响

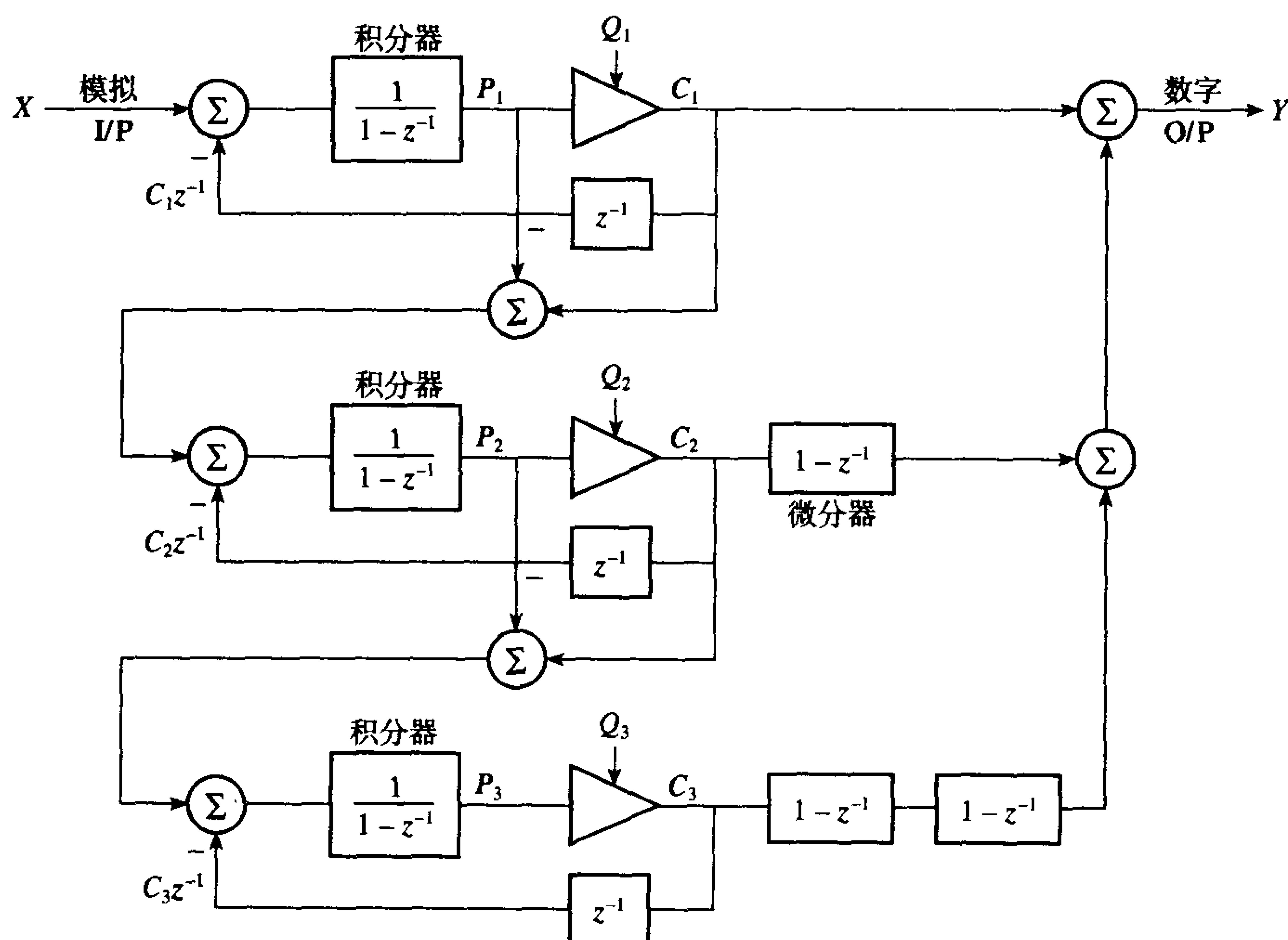


图 2.35　三阶 MASH 结构的 Σ-Δ 调制器的 z 域模型

三阶 SDM 的 MASH 结构的输出为

$$Y(z) = X(z) + E_3(z)(1 - z^{-1})^3 \tag{2.29}$$

我们注意到只有最后一级的量化噪声 $E_3(z)$影响输出，来自前两级的噪声已经被抑制。

无论 SDM 是多少阶，它的输出含有很小的带内量化噪声。而带外量化噪声很大，带外的量化噪声用低通数字滤波器滤除。由于抽样率高，直接采用数字滤波器是不实际的，而是用抽取来取代，

以达到滤波的目的，抽取也起到了将抽样率减少到希望的值的目的。滤波以后，得到的信号是B位量化了的数据。滤波起到平滑高的量化噪声的作用。典型的情况是抽取滤波器的FIR系数用16~24位表示。

图2.36给出了快速一位ADC过程的简化框图。模拟音频信号首先转换成单比特流，以3.071 MHz的速率进行Σ-Δ调制，然后采用多级抽取器（参见第9章）将单比特流下抽样到48 kHz，并且转换成16位的线性PCM字。

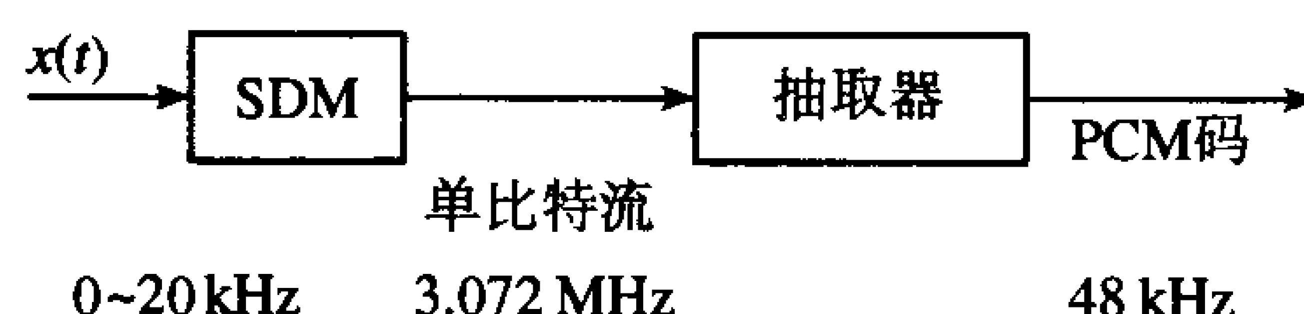


图2.36 简化的一位（过抽样）ADC

对于给定的信号类型，ADC的有限字长由通过过抽样、噪声整形和抽取达到的信噪比来确定。例如，如果我们希望16位的ADC，那么信噪比至少为96 dB。现在已经有了商业的过抽样ADC，并且可以供应现货。

例2.12 对于一个数字音频系统，其模拟音频信号输入的频率范围为0～20 kHz，采用过抽样技术和二阶Σ-Δ调制器以3.072 MHz的速率将模拟信号转换成数字比特流。Σ-Δ调制器的z域模型如图2.37所示。

(1) 解释数字比特流是如何转换成速率为48 kHz的数字多比特流的。

(2) 求由过抽样、噪声整形所带来的信号量化噪声比的总的改善，即估计数字转换器的有效分辨率，用位表示。

解：

(1) 通过抽取将单比特流转换成多比特流（降低抽样过程）。SDM的输出带内噪声很小，而带外有很大的噪声，带外噪声由低通数字滤波器滤除。由于抽样率高，直接使用数字滤波器是不实际的，而是用抽取来取代，从而达到滤波的目的。抽取也起到将抽样率减少到希望值的目的。典型的情况是采用二级抽取器（16和4的因子）。滤波以后，得到的信号是B位量化后的数据。滤波起到平滑高的量化噪声的作用。典型的情况是抽取滤波器的FIR系数用16～24位表示。

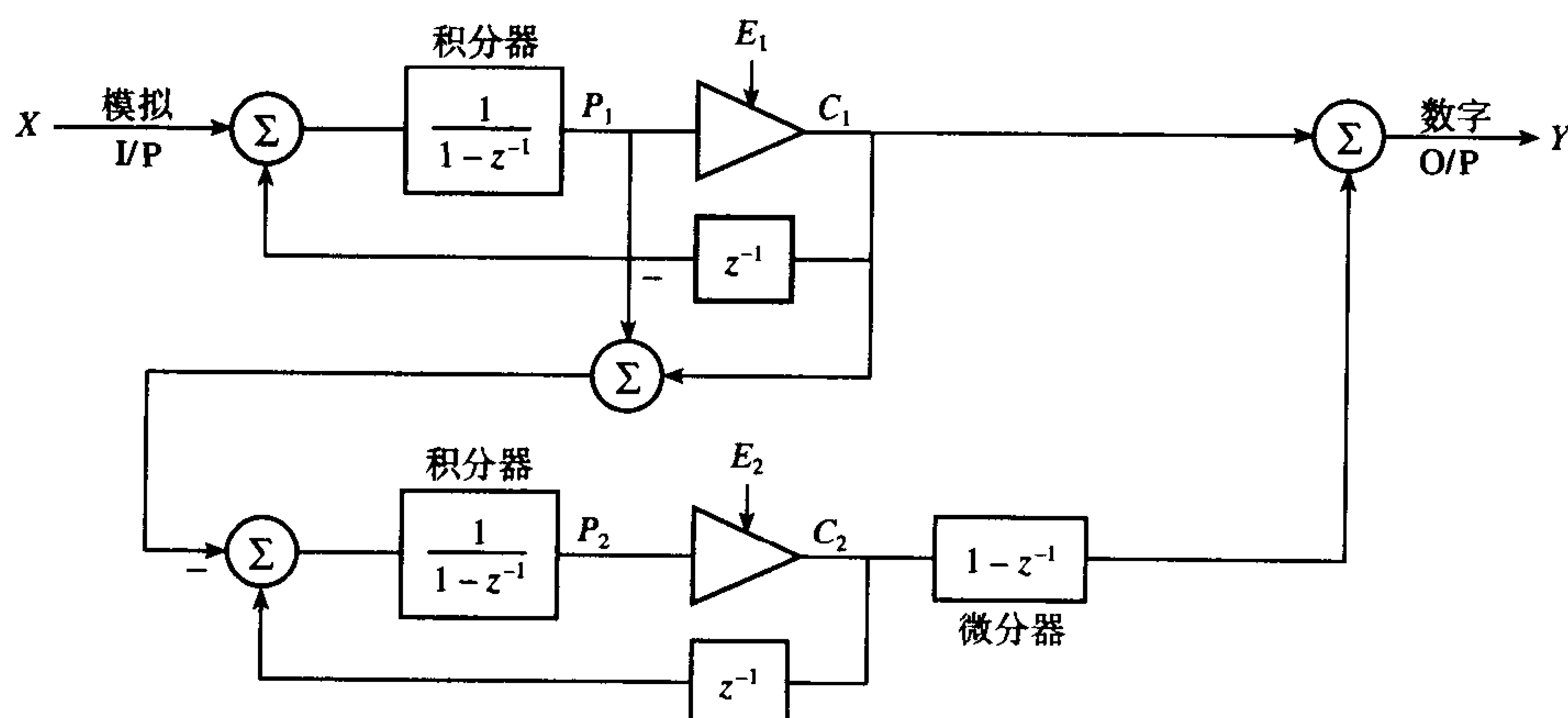


图2.37 二阶SDM的z域模型（例2.12）

(2) 通过如下的简化分析可以得到有效分辨率的估计。

通过过抽样和噪声整形可以减少噪声功率，由于过抽样而引起的噪声功率减少由过抽样比给出。

过抽样比为

$$\frac{F_s}{2f_{max}} = \frac{3.072\times10^6}{2\times24\times10^3} = 64$$

即量化噪声功率减少了 18 dB。

根据 Σ-Δ 调制器的 z 域模型，量化噪声的传递函数为

$$N(z) = (1 - z^{-1})^2$$

这是一个高通滤波器，它在 dc 处有一个双重零点，它衰减低频端的噪声分量，如图 2.38 所示。幅度响应为

$$|N(z)|^2_{z=e^{j\omega T}} = |(1 - e^{-j\omega T})^2|^2$$

对于 $f = 24$ kHz（带沿）、$F_s = 3.072$ MHz，$\omega T = 2.8125°$，以及

$$|N(e^{j\omega T})|^2 = 2.412\times10^{-3}$$

SQNR 减少了 52.35 dB，ADC 的有限字长主要由通过过抽样和噪声整形获得的信噪比来确定，总的 SQNR 减少了 70.41 dB，这对应于 11.4 位的 AD 有效分辨率（根据 SQNR = $6.02B + 1.77$ dB）。

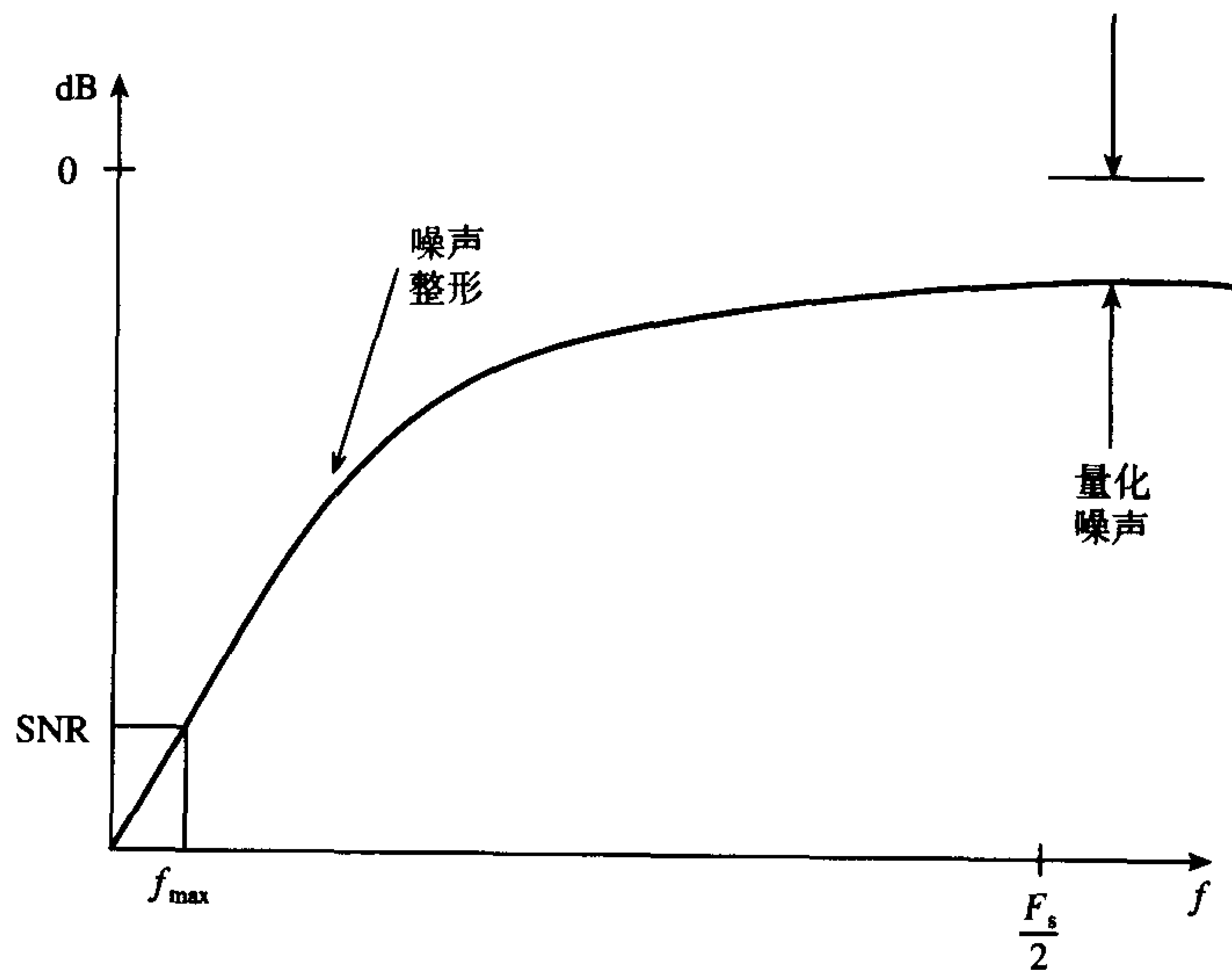


图 2.38　噪声整形对量化噪声的影响

2.6　数模转换过程：信号恢复

数模转换过程是用来在经过数字处理、发射和保存后将数字信号转换成模拟信号。之所以这样转换，可以是为了生成声频信号来驱动扬声器（像激光唱盘那样），或者发出报警的声音。最常见的形式如图 2.39 所示，从图中可以看出，它主要由两个部件构成：DAC（数字模拟转换器）和低通滤波器，低通滤波器有时也称为重构、平滑或抗像频滤波器。

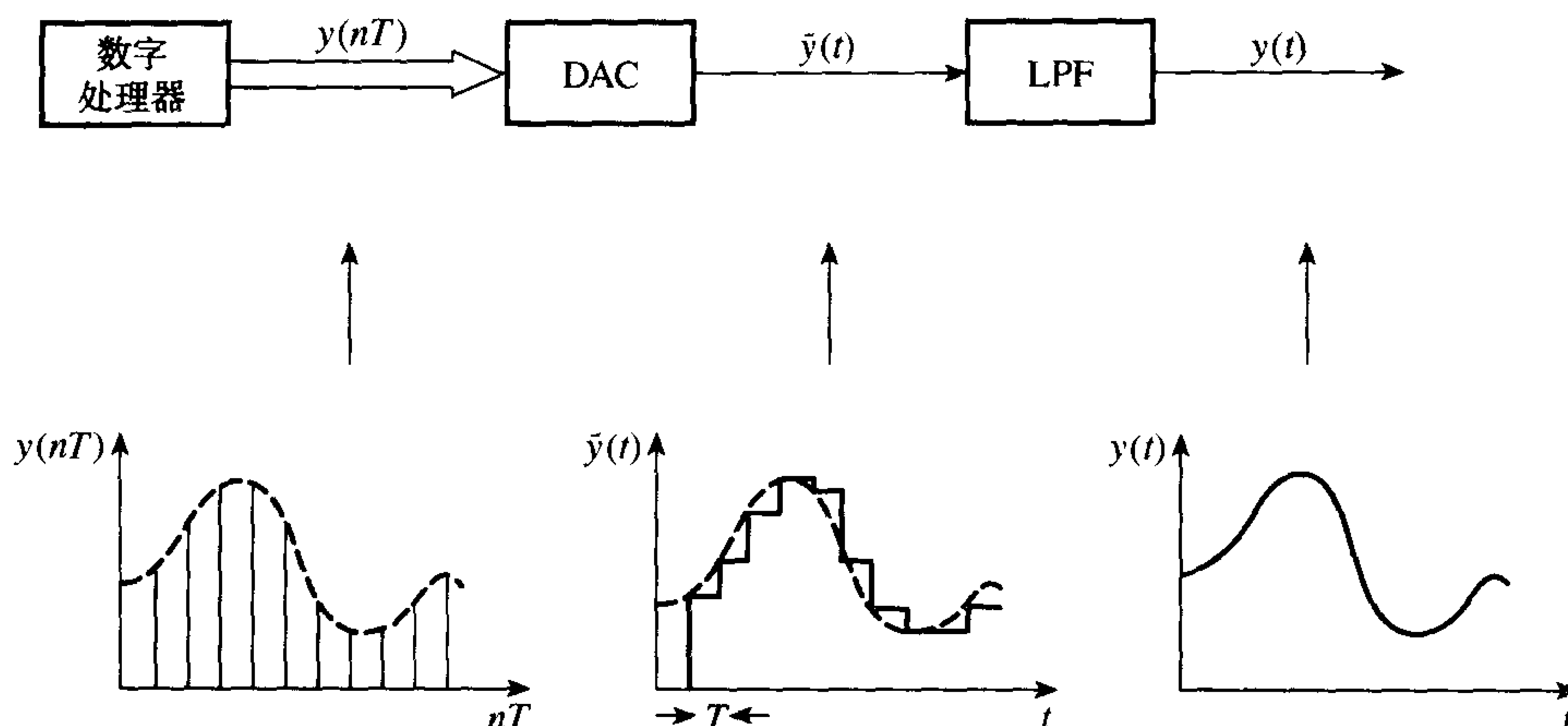


图 2.39　用来在数字处理后恢复模拟信号的数模转换过程。注意，输入到DAC的是一串脉冲，而DAC的输出是阶梯型的，因为每个脉冲保持时间 T(s)

2.7　DAC

基本的DAC并行接收数字数据，产生与输入数字代码有关的模拟输出信号。一个寄存器用来缓冲DAC的输入，确保它的输出保持相同，直到DAC流入下一组数字输入。寄存器可能在DAC的外部，也可能是DAC芯片的一部分，如图2.39所示。在某些应用中，要求附加电路来防止虚假数字代码在DAC的输出中产生瞬时毛刺（transient spike）。

图中所示的DAC称为零阶保持，通过比较它的输出 $\bar{y}(t)$ 和输入 $y(nT)$，很明显对于流入DAC的数字代码，它的输出保持一段时间 T，结果使得在DAC的输出为阶梯形状。在频域中，DAC的保持行为引入了一种称为 $\sin x/x$ 的失真或孔径失真，其中 $x=\omega T/2$。

图 2.40 给出了零阶保持DAC在输入和输出端信号的时域和频域的表示，下面几点要注意：

- DAC的输入和输出信号都是宽带的信号，每个信号都是由信号频谱（已经被数字化）加上无穷多个原始信号频谱的像，这些频谱的像中心位于抽样频率的倍数处。
- 输出信号频谱的幅度被乘上了 $\sin x/x$ 函数，它的作用像低通滤波器，对像频有大的衰减。$\sin x/x$效应是由于DAC的保持行为，在信号恢复中引入了幅度失真。在某个给定的频率处，由于这种效应引起的平均误差表示为偏离 1 的百分比：

$$(1-\sin x/x)\times 100\% \tag{2.30}$$

对于零阶保持，函数 $\sin x/x$ 在半抽样频率处下降了大约 4 dB，在该频率处给出了大约 36.4% 的平均误差。利用均衡可以消除孔径误差。在实际中是这样实现的：在将信号转换成模拟信号之前，先让它通过幅频响应为 $\sin x/x$ 形状的数字滤波器。

在某些应用中，在实际的抽样点加到DAC时，用数字信号处理器在抽样点之间插入或内插一些点，这有助于平滑模拟信号，得到比简单的零阶保持器更好的结果。另一种方法就是以比抽样定理要求高得多的抽样率，采用多速率技术来进行模数转换（Goedhart et al., 1982），它具有改善信噪比性能以及简化抗像频滤波器的优点。这种方法在要求高品质的音频信号的地方极为流行，进一步的讨论在 2.9 节和第 9 章中给出。

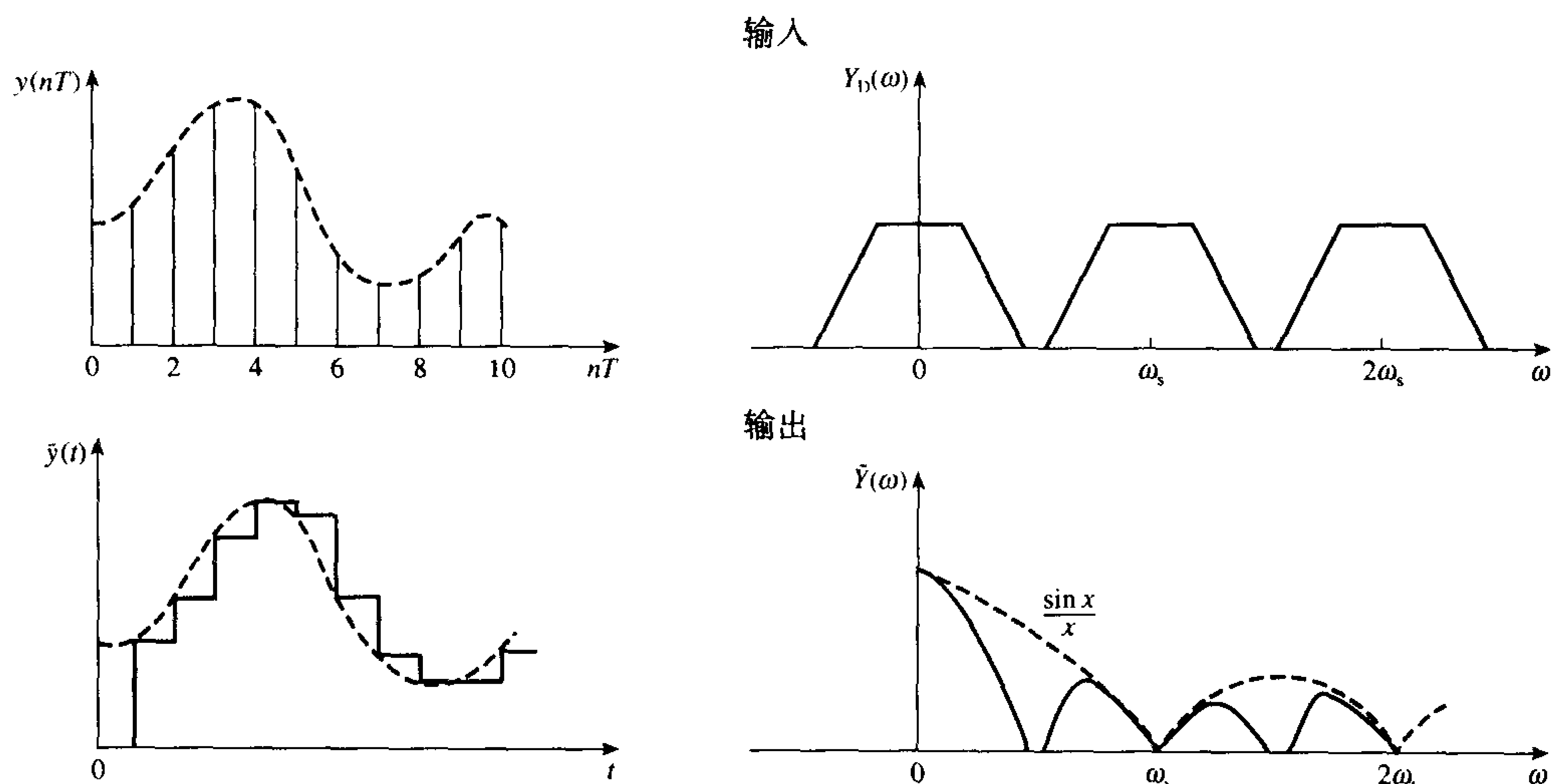


图2.40　DAC输入和输出信号的时域和频域表示。注意sin *x*/*x*的主要作用是作为一个低通滤波器

2.8　抗镜像滤波

DAC的输出包含有不需要的高频分量和像频分量，在期望的频率分量和抽样（即更新）频率的整数倍为中心的地方都存在像频分量。高频分量可能引起不希望的旁瓣效应，这与具体的应用有关。例如，在CD播放器中，尽管像频听不到，但它可能引起播放器的放大器过载，以及与希望的基带频率分量产生交叉调制（intermodulation），其结果是造成音频信号质量无法接受的恶化。

输出（即抗镜像）滤波器的作用就是通过消除不希望的高频分量来平滑DAC输出中的阶梯形信号。输出滤波器衰减的要求取决于模拟信号对后续模拟电路的影响。一般来说，抗镜像滤波器的要求类似于抗混叠滤波器的要求。

2.9　D/A转换中的过抽样

过抽样DAC的动机类似于过抽样ADC。在过抽样DAC中，数据转换成模拟信号的抽样速率增加了许多倍（例如64倍），目的是为了以非常精细的间隔产生模拟信号的抽样值。这样，只需要相对简单的模拟抗像频滤波器来平滑或者消除带外噪声。以非常高的速率抽样，产生带宽超过从零到抽样频率一半的带宽的信号。量化噪声功率均匀地散布在这个较宽的频带内，使得用低分辨率DAC达到高分辨率D/A转换的效果。

如同在ADC中的情况，靠过抽样自身达到希望的DAC分辨率是不合适的，因此噪声整形是必需的。所以，实际的过抽样DAC一般由四部分组成：过抽样数字滤波器，噪声整形器（例如Σ-Δ调制器）、低分辨率DAC（例如一位DAC）和简单的模拟抗镜像滤波器，参见图2.41。

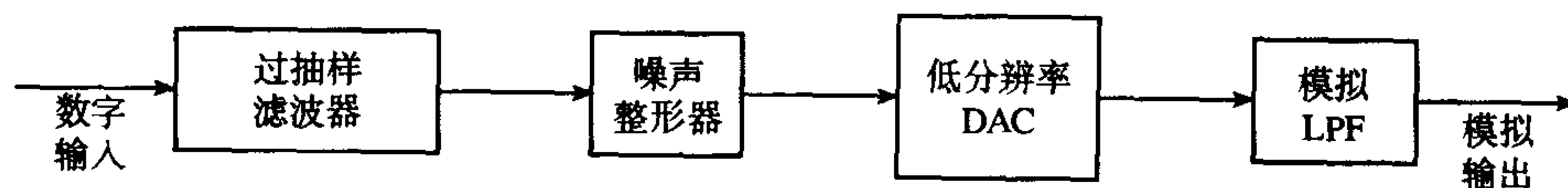


图2.41　过抽样D/A转换的原理框图

过抽样滤波器用来增加抽样率、衰减像频分量。减少字长（例如从16位减少到1位）将增加很多量化噪声，由于抽样率增加了很多，所以这些量化噪声散布在很宽的带宽内。此外，噪声整形器将量化噪声的大部分移到了信号带宽之外的较高频带。使用简单的抗镜像滤波器来平滑带外噪声。

2.9.1 CD播放器中的过抽样D/A转换

我们将通过考虑CD播放器是如何实现的来说明过抽样D/A转换的原理，参见图2.42。在解码和误差校正以后，从CD中读出的数字信号是用16位表示的字，表示44.1 kHz速率的音频信息。如果将数字代码直接转换成模拟信号，将会以抽样频率44.1 kHz的整数倍为中心产生像频频带（参见图2.43(a)）。尽管像频在20 kHz以上是听不见的，但是，它们通过播放器的放大器和扬声器时可能会引起过载，或者可能引起交叉调制失真。因此，基带以上的频率分量需要衰减至少50 dB，能够提供这一级别衰减的模拟滤波器必须满足很高的指标要求，并且要求可调来保证两个立体声道的滤波器是匹配的。

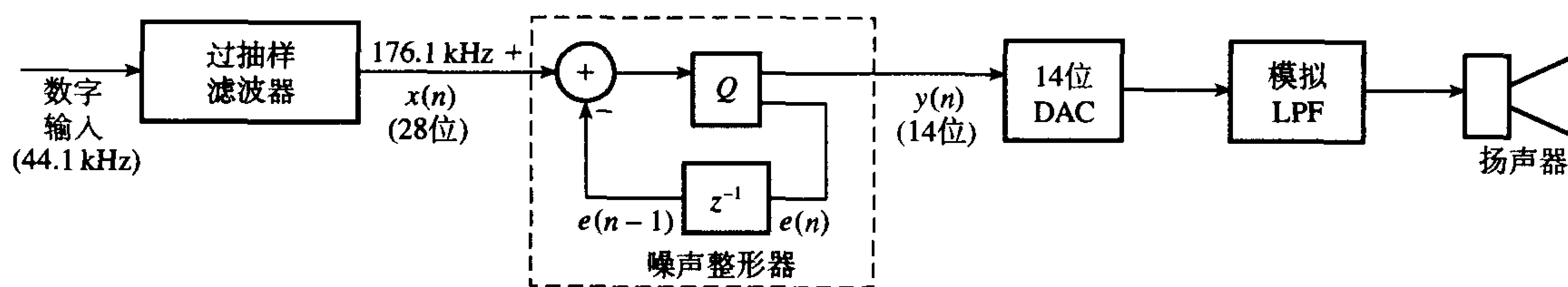


图2.42 在具有四倍过抽样和噪声整形的CD播放器中音频信号重现的简化框图

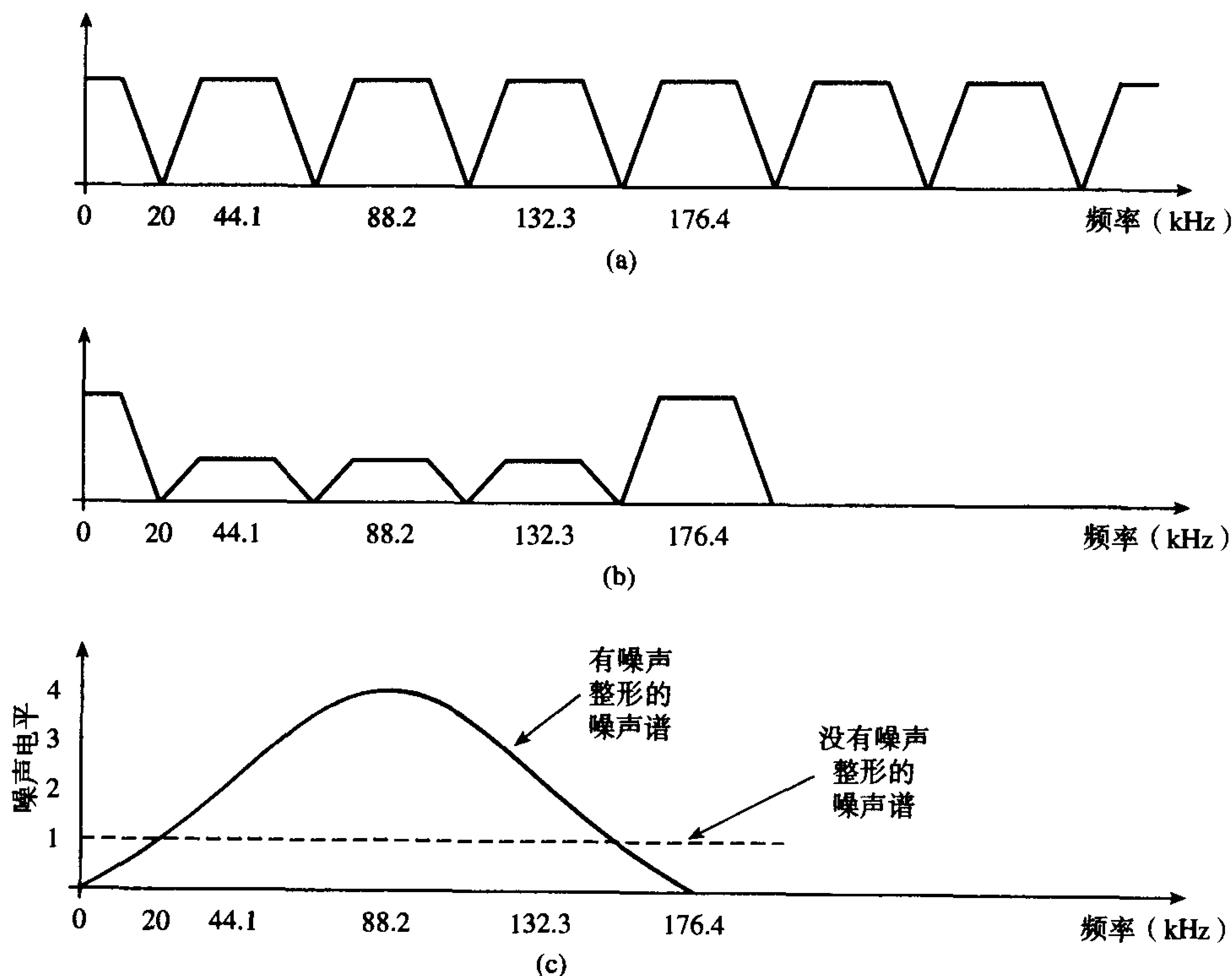


图2.43 (a) 以44.1 kHz抽样的音频信号抽样值的频谱；(b) 四倍过抽样的音频信号频谱，频谱图表明像频分量减少；(c) 噪声整形器对噪声频谱的影响。我们注意到，在音频带（0~20 kHz），噪声电平要小于高频的噪声电平，并且也比没有噪声整形的噪声电平小（虚线）

为了避免这样的问题，在CD播放器中采用过抽样滤波器，在D/A转换前将数据的抽样频率乘4得176.4 kHz（4 × 44.1 kHz）就可以实现。图2.43(b)给出了4倍过抽样和数字滤波后信号的频谱。可以看出20 kHz以上的像频分量在过抽样后大为减少，使得它很容易滤除。在时域，这种效应将产生非常好的阶梯信号；在频域，像频分量现在被移到更高的频带。

过抽样滤波器（28 位字）的输出加到噪声整形器，然后通过舍入被量化成 14 位字，参见图 2.42。（过抽样滤波器的系数字长为 12 位，而它的输入由 16 位字组成。滤波后，输出由 28 位字组成。）量化误差被反馈，并且与过抽样滤波器的输出组合。噪声整形器起到将量化噪声移到高频端的作用。

过抽样、滤波和噪声整形的组合效应将大大减少像频分量和信号频带内的量化电平，这使得我们用 14 位 DAC 仍然可以达到 16 位 DAC 等效的 SNR 性能。可以证明，4 倍过抽样滤波器和噪声整形器可以分别提供 6 dB 和 7 dB SNR 的改善。

DAC 的保持效应在频谱产生 $\sin x/x$ 效应，它可以进一步减少像频分量，那么就可以用简单的抗镜像滤波器来恢复音频信号。

例 2.13　图 2.44(a)描述了一种恢复模拟信号的结构，信号已经在某个实时数字音频系统中进行了数字处理。模拟信号的基带从 dc 到 20 kHz，数模转换器以 176.4 kHz 的速率更新，像频至少抑制 50 dB，感兴趣的信号分量最大有 0.5 dB 的变化。求抗镜像滤波器的最小阶数和截止频率，假定抗镜像滤波器具有巴特沃斯特性，阐述所做的任何合理假设。

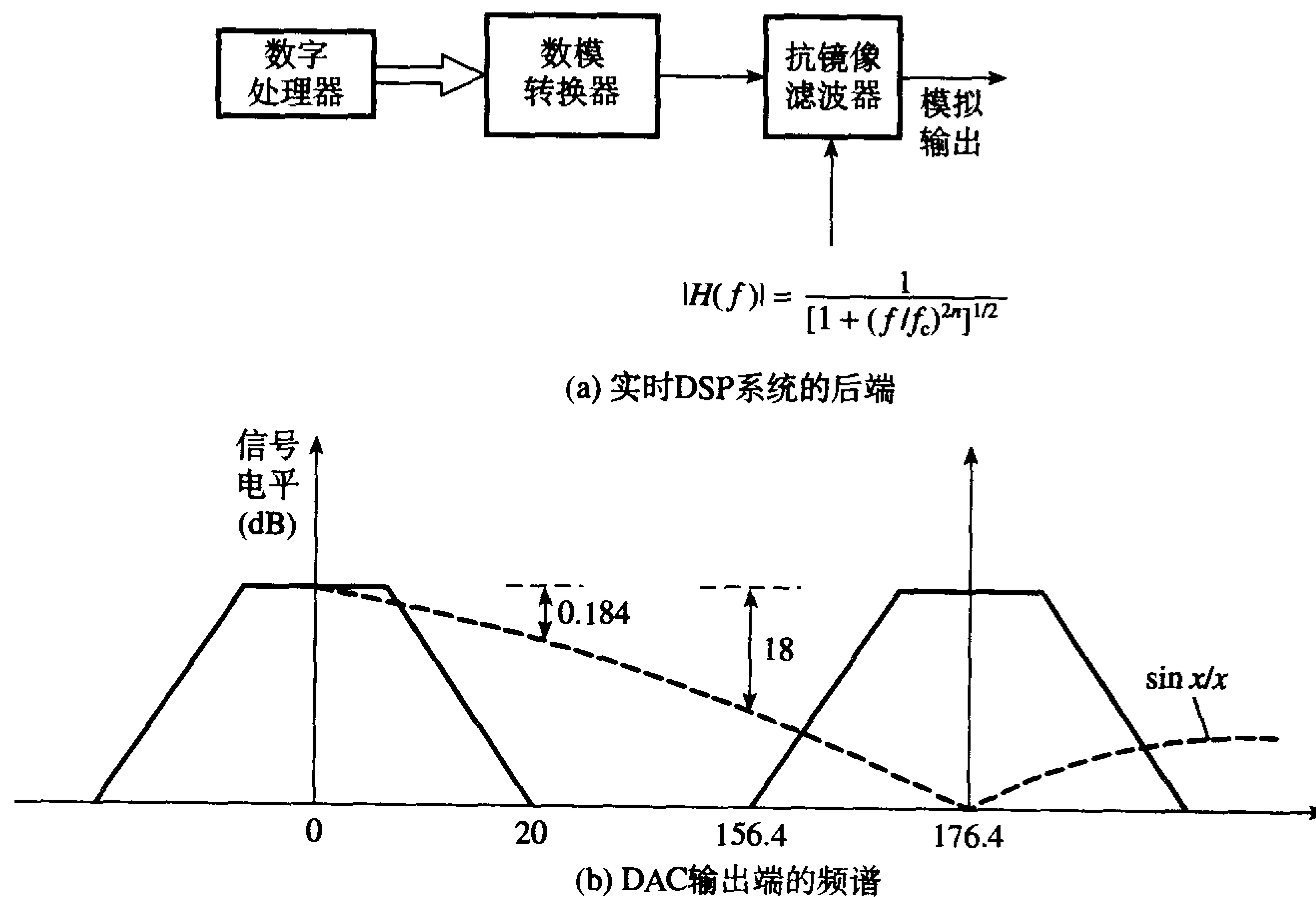


图 2.44　恢复模拟信号

解：

假定零阶保持，DAC 输出端的频谱是信号频谱与 $\sin x/x$ 响应的乘积，参见图 2.44(b)，在两个边界频率点 20 kHz 和 156.4 kHz，$\sin x/x$ 对信号频谱的衰减如下：

在 20 kHz 处：$\sin x/x = 0.9789$（用 $x = \omega T/2$）$\rightarrow -0.184$ dB

在 156.4 kHz 处：$\sin x/x = 0.125 \rightarrow -18$ dB

因此，在通带内，输出滤波器不会有超过 0.5−0.184 = 0.316 dB 的偏差，在阻带内，至少要求附加 50−18 = 32 dB 的衰减，因此，

$$20 \log [1 + (20/f_c)^{2n}]^{1/2} \leqslant 0.316 \text{ dB}$$

$$20 \log [1 + (156.4/f_c)^{2n}]^{1/2} \geqslant 32 \text{ dB}$$

求解 n 得 $n = 2.4 \simeq 3$（整数），$f_c = 30.76$ kHz。

2.10 具有模拟输入/模拟输出信号的实时信号处理的限制

我们已经讨论了实时DSP系统中由模数和数模转换带来的限制和误差，现在我们概括一下这些限制及其可能的解。

- 在表示数据时采用有限位数引入固有的误差，即量化误差，这种误差会传播到随后的信号处理。处理这种误差有两种方法：增加ADC的分辨率；过抽样信号，用进一步的DSP来改善SNR（更详细的内容参见第9章）。
- 高分辨率ADC和DAC一般是慢速的（除非很昂贵的转换器）。一般来说，ADC转换一个模拟的抽样值要几微秒，DAC要花几分之一微秒来固定。这种延迟限制了可达到的最大抽样频率。事实上，对于当前的技术来说，在大多数实时DSP中，ADC/DAC是主要的瓶颈。
- ADC/DAC易受到许多其他误差的影响，包括温度效应和非线性。因此，带模拟输入的、好的实时DSP系统应该有高质量的模拟输入和输出部分。
- 抽样保持的输出是宽带（因为像频），并且会增加ADC输入端的噪声。
- 来自感兴趣频带外信号能量的混叠误差总是存在的。为了将混叠减少到可接受的电平，在抽样前要带限信号，或者有可能采用过抽样。
- 零阶DAC的使用将引入sin x/x效应，将逐步减少信号的高频分量，这可以通过采用频率响应为$x/\sin x$的数字滤波器来补偿。
- 抗混叠滤波器引入误差，通常它们是幅度和相位误差，这些滤波器的幅度响应在感兴趣的频带内不是平的。幅频特性好的模拟滤波器，相位特性固有地差，这意味着信号之间的谐波关系是失真的。在多通道系统中，问题是多方面的，由模拟信号调节器引入的失真对每一个通道是不同的，可能需要补偿。
- 抽样保持误差包括采集时间、孔径不确定、转换期间的下落误差（droop error）、在保持模式中的直通等。
- 在现代DSP系统中，特别是在像CD播放器那样的数字音频系统中，其趋势是用一位ADC和DAC，这些新的器件利用了多速率技术的优势（参见第9章）。

2.11 应用例子

模拟I/O接口技术的应用在大多数实时DSP系统中是相当普遍的，新的应用例子利用了抽样定理以及与抽样有关的因子（称为抽样率），抽样后数据的频谱以抽样频率的倍数重复这样的事实，有效信号的有限带宽（即实际信号分量被限制在0到$F_s/2$之间），对量化噪声抽样的效应，等等。这就导致了低成本、高分辨率A/D和D/A转换器的开发（参考2.5节和2.9节），这些应用依赖于多抽样率技术，该技术将在第9章讨论，因此这些应用也放到第9章讨论。

在通信中，用带通抽样技术来加强接收机的设计。在这些应用中涉及到的问题要求很好地理解多速率，所以，我们也将这些讨论放到后面进行。

2.12 小结

一般认为，实时DSP系统是模数转换部分、数字处理器和数模转换部分组成的。在这样的系统中，模拟和数字之间转换要求的时间限制了DSP系统能够处理的最大信号带宽。在转换过程中

用到的器件可能引入相当大的误差或者信号的损失，大多数的误差都可以通过仔细选择器件（ADC和DAC等）和系统参数（抽样频率等）使其达到最小。例如，以足够高的频率进行抽样以及采用合适的带限滤波器可以减少混叠。

习题

抽样和混叠控制

2.1 如何理解下列术语：

(a) 奈奎斯特频率？

(b) 奈奎斯特率？

(c) 抽样率？

(d) 抽样频率？

2.2 信号具有图 2.45 描述的频谱，求避免混叠的最小抽样频率。假定信号以 16 kHz 的速率进行抽样，画出抽样的信号在间隔 ± 16 kHz 上的频谱。在图中标出包括折叠频率的有关频率。

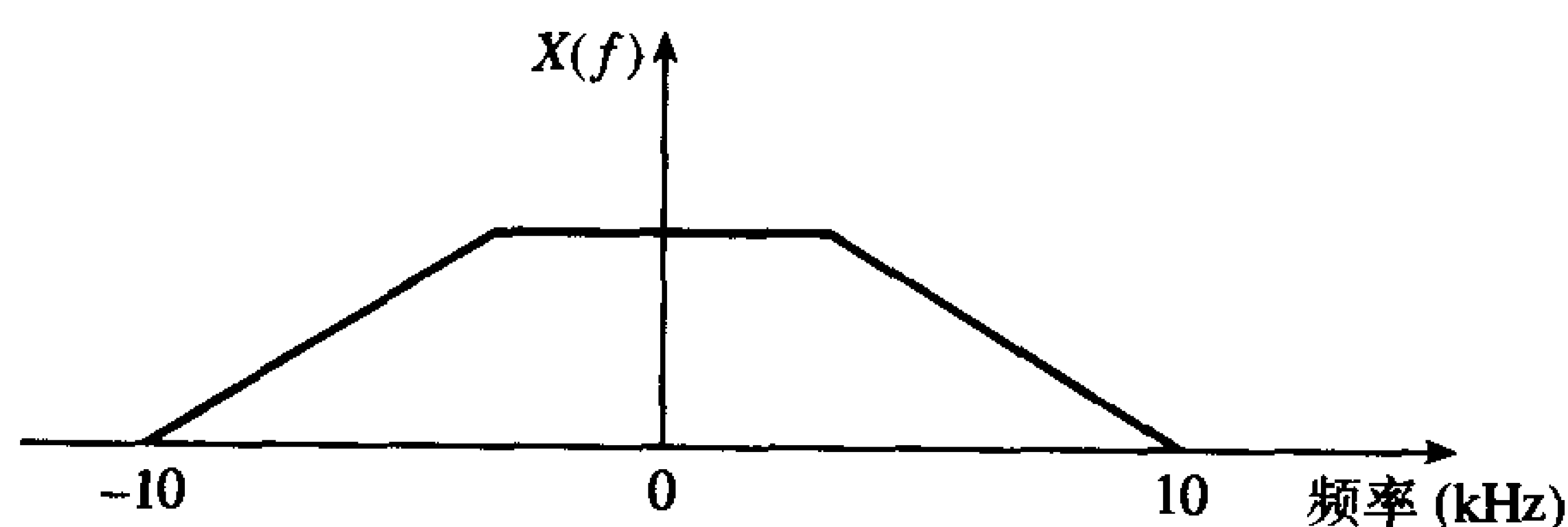


图 2.45

2.3 解释为什么只考虑抽样定理并不足以确定在实际的 DSP 系统中使用的抽样频率？

2.4 解释实时 DSP 系统的抗混叠滤波器和抗镜像滤波器，为什么两种滤波器的要求在 DSP 系统中常常是相同的？

2.5 在某个实时 DSP 系统中，模拟输入部分的要求是

感兴趣的频带	0 ~ 4 kHz
最大允许通带波纹	≤ 0.5 dB
阻带衰减	≥ 50 dB

求具有巴特沃斯特性的抗混叠滤波器的最小阶数以及满足要求的合适的抽样频率。

2.6 实时 DSP 系统的模拟输入用双极型的 16 位 ADC 进行数字化，输入信号的峰峰幅度为 ± 10 V，输入信号的频谱在 0 ~ 10 kHz 频带内，估计最小的：

(1) 抗混叠滤波器的阻带衰减 $A_{\min}$；

(2) 抽样频率 F_s，要使得通带内的混叠误差刚好在量化噪声电平之下（假定抗混叠滤波器采用六阶巴特沃斯滤波器）。

2.7 一个具有均匀功率谱密度的模拟信号，用具有下列幅频特性的滤波器变成带限信号：

$$|H(f)| = \frac{1}{[1 + (f/f_c)^6]^{1/2}}$$

其中 f_c = 3.4 kHz，信号用线性的 8 位 ADC 数字化，求最小抽样频率，要求在通带内的混叠误差小于量化误差电平。

2.8 加到实时DSP系统的模拟信号在数字化前用具有三阶巴特沃斯特性的模拟滤波器带限到30 Hz。如果由于抽样而引起的混叠误差小于通带内信号电平的1%，求系统所要求的最小抽样频率F_s。

如果信号在数字化和处理后又转回到模拟信号，在30 Hz处由孔径效应引入的平均误差是多少？假定输入信号用理想的抽样器和ADC数字化，但是用零阶保持DAC，在输入和输出假定一个共同的抽样频率256 Hz。

2.9 图2.46描述一个实时DSP系统的前端，假定输入为宽带信号，

(a) 画出在$\pm F_s/2$频率范围内抽样前（A点）后（B点）信号的频谱；

(b) 求在15 kHz和奈奎斯特频率（即30 kHz）处信号和混叠误差电平；

(c) 求最小抽样频率$F_s(\min)$，给定15 kHz处的信号与混叠误差电平之比为10 : 1。阐述任何所做的假定。

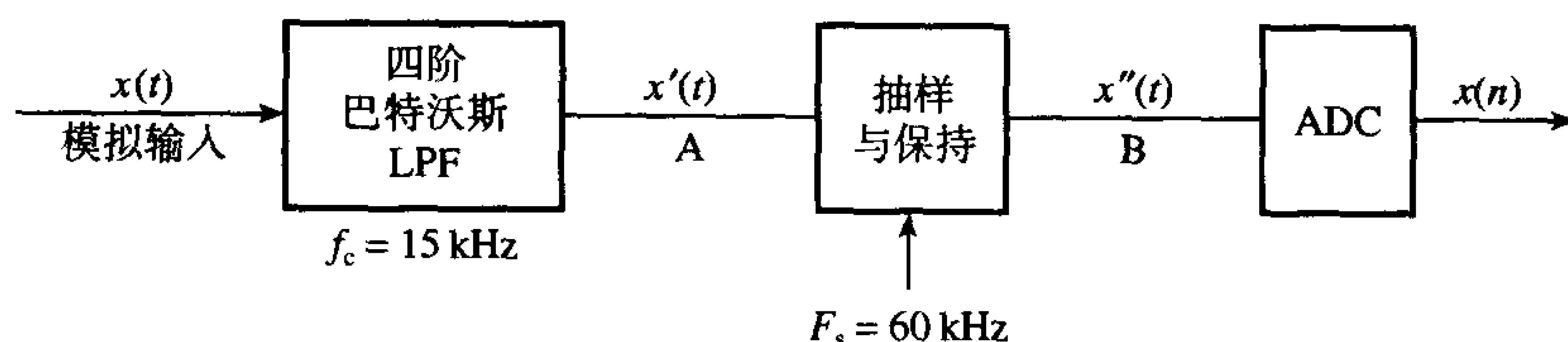


图2.46 一个实时DSP系统的前端

2.10 图2.47描述了一个实时DSP系统，假定感兴趣的频带扩展到0 ~ 100 Hz，采用16位双极性ADC，

(1) 估算抗混叠滤波器的最小阻带衰减$A_{\min}$；

(2) 估算最小抽样频率F_s；

(3) 对于估算出来的$A_{\min}$和F_s，估算阻带内相对于信号电平的混叠误差电平。

画出并标注模拟滤波器输出端信号的频谱以及抽样后信号的频谱，假定在输入端信号为宽带的。

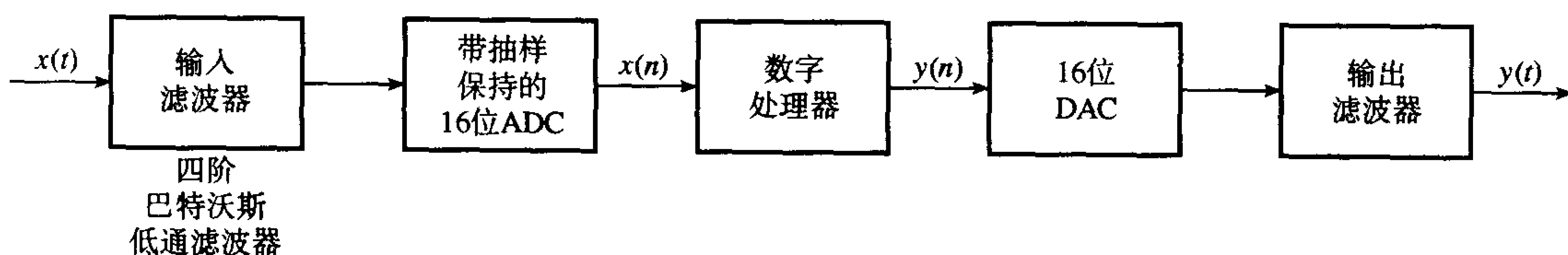


图2.47 实时数字信号处理系统

2.11 (a) 简要讨论确定实际DSP系统中混叠误差的三个主要因素，指出混叠控制时这些因素之间的相互关系。

(b) 一个具有均匀功率谱密度的模拟信号用具有下列幅频特性的抗混叠滤波器变成带限信号：

$$\frac{1}{\sqrt{1+\left(\frac{f}{f_c}\right)^8}}$$

其中$f_c = 40$ Hz，信号用线性的12位ADC数字化，求：

(1) 最小抽样频率，要求在通带内的混叠误差不大于量化误差电平；

(2) 相对于 ADC 量化噪声水平的最大阻带信号电平，用 dB 表示。

阐述任何所做的合理假设。

(c) 写出带通抽样定理方程，解释为什么定理在数字通信系统中有用。

(d) 从(c)的方程开始，推导带通信号理论上最小抽样率的表达式。假定信号上沿频率与带宽之比是整数，解释为什么理论上的最小抽样率在实际中是不能用的？

带通欠抽样

2.12 图 2.48(a)描述了一个多信道通信系统的前端，接收信号频谱如图 2.48(b)所示，图中指出了信道号。在信号以尽可能最低的速率数字化前，用带通滤波器在希望的信道中分离出信号。

假定理想的带通滤波器具有下列特征：

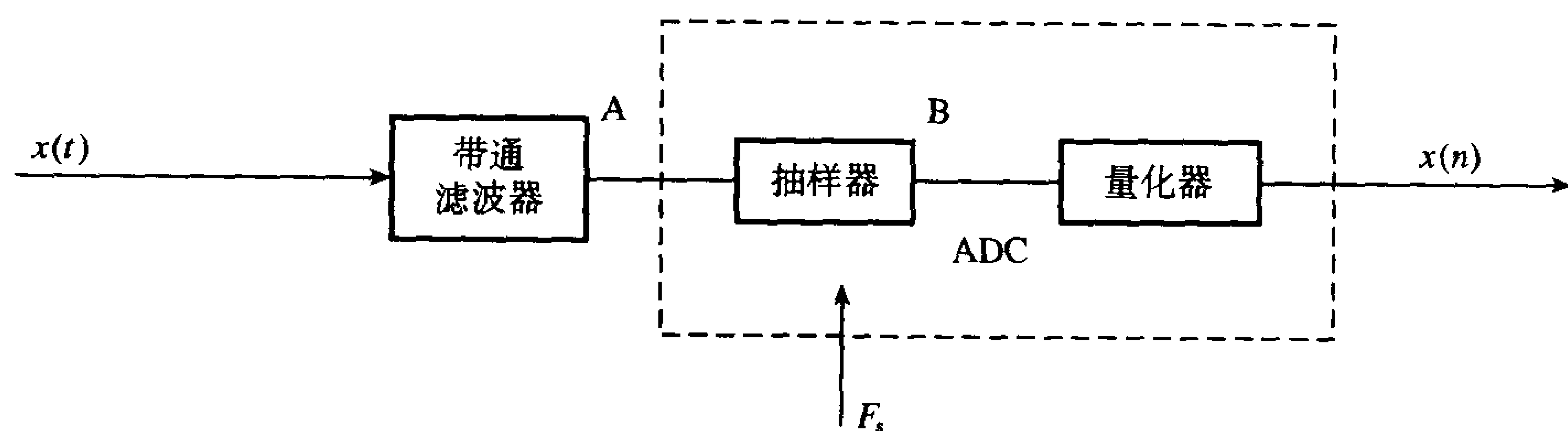
$H(f) = 1$　如果 $10\ \text{kHz} \leqslant f \leqslant 20\ \text{kHz}$

　　　0　其他

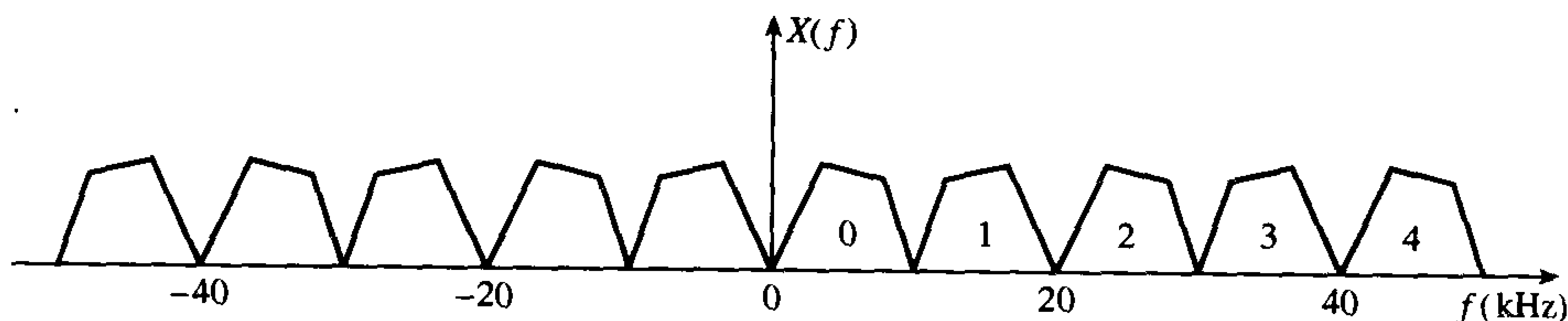
(a) (i) 求最小理论上的抽样频率；

(ii) 画出抽样前（A 点）后（B 点）信号的频谱。

(b) 对于通过信道 2 的带通滤波器，重做(i)、(ii)。



(a) 系统的前端



(b) 接收信号的频谱

图 2.48　多通道系统的前端和接收信号的频谱

2.13 图 2.49 画出了一个窄带信号的频谱，对于下列三种情况，求出并画出在 $\pm F_s/2$ 范围内抽样信号的频谱：

(1) $\frac{f_H}{B} = 3$;

(2) $\frac{f_H}{B} = 4$

(3) $\frac{f_H}{B} = 4.5$

假定信号的带宽 $B = 5$ kHz，在每种情况下信号以 $2B$ 的速率抽样。

2.14 (a) 对于一个频率分量在范围 $20\ \text{MHz} < f < 30\ \text{MHz}$ 内的带通信号，求避免混叠的最小理论抽样率 F_s。证明你的答案，并且解释为什么最小理论抽样率在实际中并不采用。

(b) 假定(a)中的带通信号的频谱如图2.50所示，求抽样后信号在$\pm 2F_s$间隔上信号的所有频带（包括像频分量）的边沿频率。在图形纸上画出并清楚地标注出在该间隔上抽样后信号的频谱。

(c) 如果模拟带通信号在任一带沿增加5 kHz的防护频带，计算避免混叠的抽样率的可允许范围。

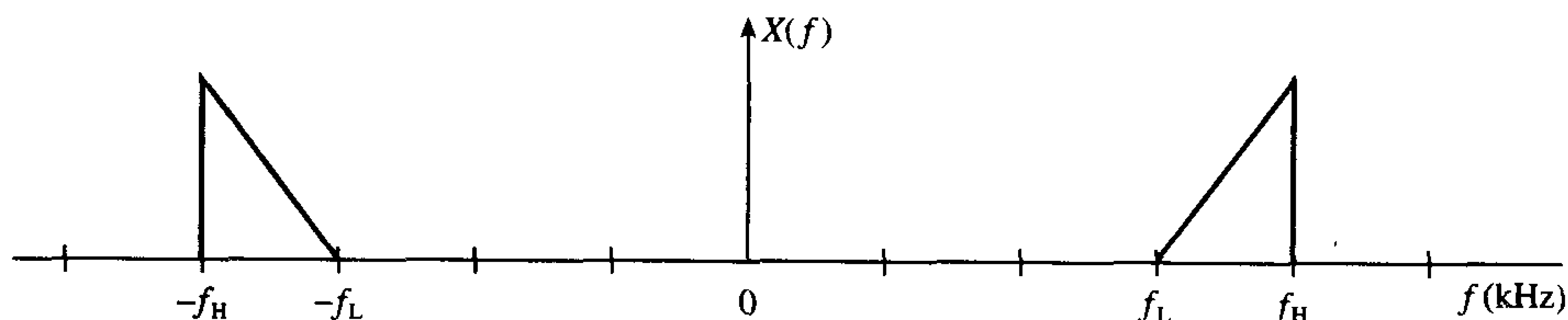


图2.49　窄带信号的频谱

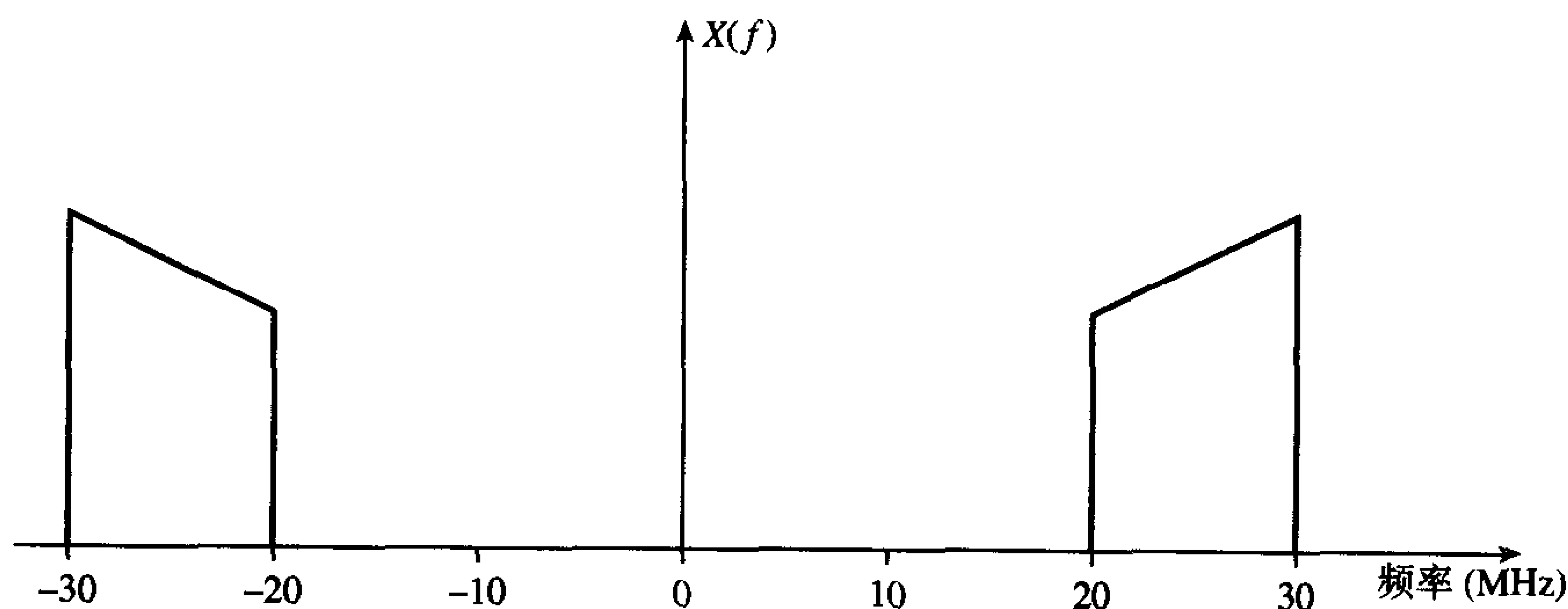


图2.50　带通信号的频谱

2.15 (a) 简要说明带通欠抽样技术的原理，解释该项技术在实际中的益处。

(b) 数字无线接收机在第二级采用50 kHz的IF（中频），

(i) 如果中频信号带宽为6 kHz，求系统避免混叠的最小抽样频率F_s；

(ii) 画出并标识抽样后信号在$\pm F_s$上的频谱，解释你是如何得到抽样信号频谱的，并解释它的形状。

假定采用整数带抽样技术，且在第二IF级的信号频谱如图2.51所示。

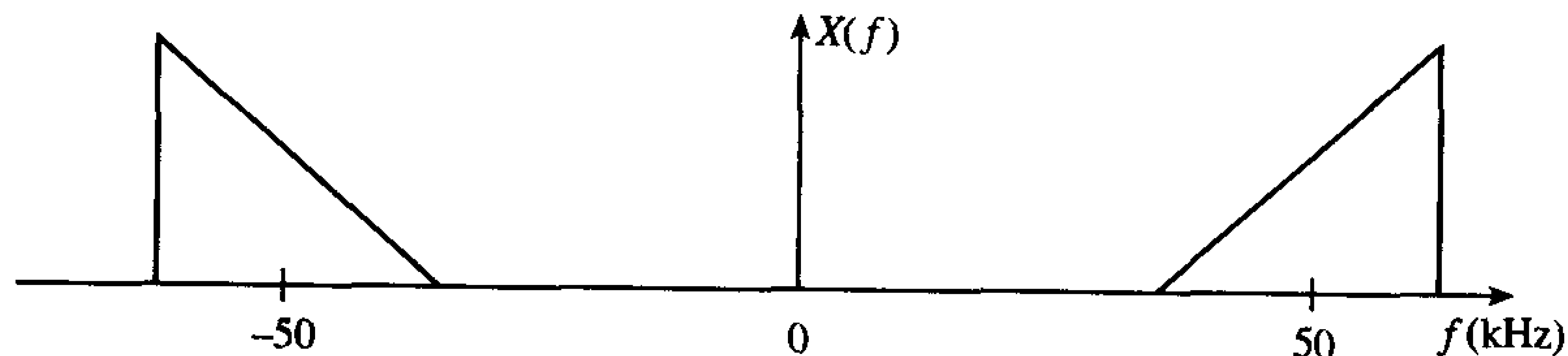


图2.51　第二IF级的信号频谱

2.16 (a) 图2.52(a)给出了数字接收机第二IF级的频谱，其中中频是2.976 MHz，用图解法证明：IF信号可以按128 kHz的速率抽样而不产生混叠。

(b) 借助图解法证明：如果IF频率是3 MHz，如果IF信号以128 kHz的速率进行抽样，那么就会有混叠的输出。

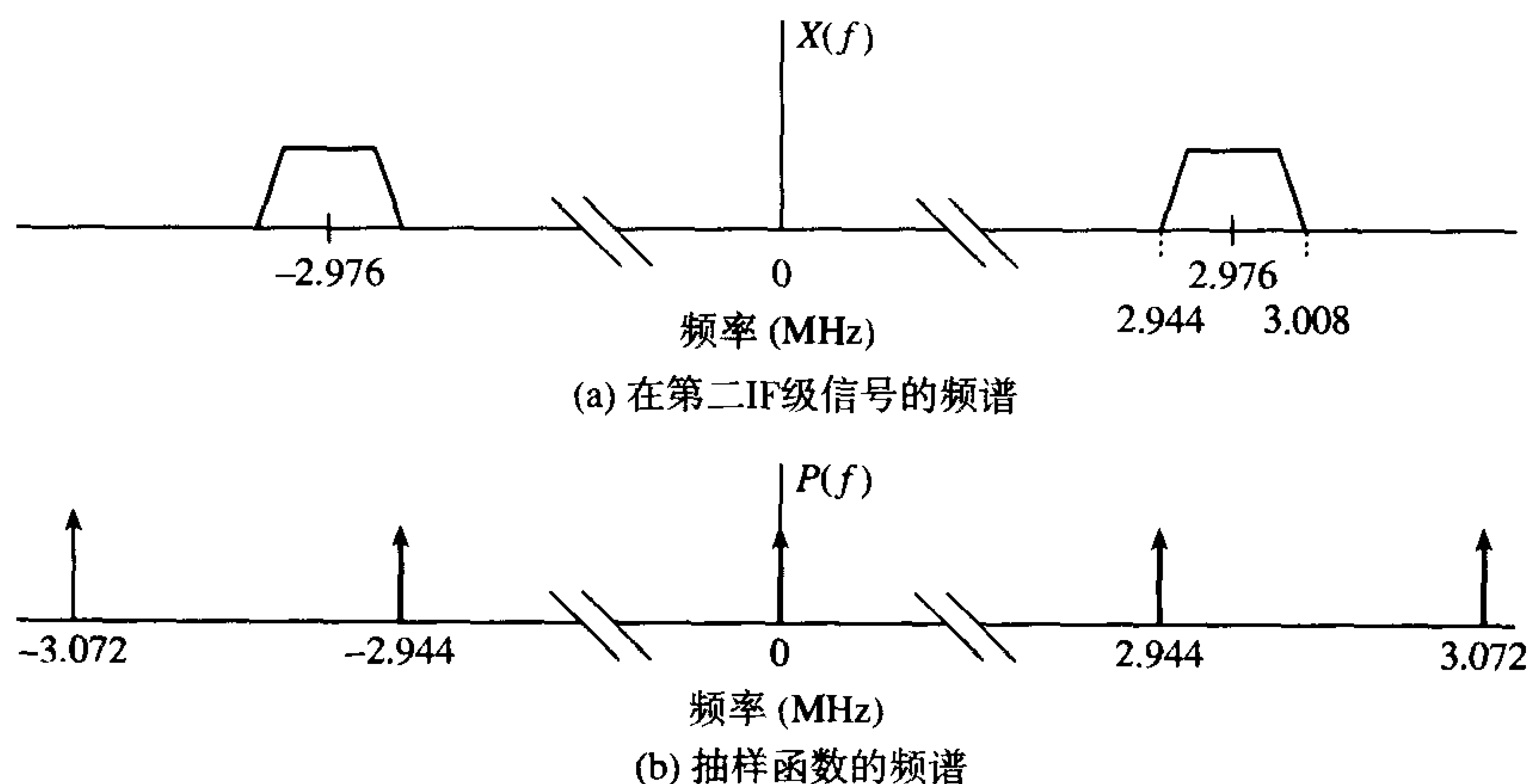


图 2.52　习题 2.16 的示意图

2.17 (a) 写出带通抽样定理方程，解释为什么带通抽样定理在数字通信系统中有用。

(b) 从方程开始，推导带通信号理论上最小抽样率的表达式。假定信号上沿频率与带宽之比是整数，解释为什么理论上的最小抽样率在实际中是不能用的？

A/D 转换中的量化噪声

2.18 一实时 DSP 系统使用了 16 位线性双极型 ADC，输入信号范围为 ±5 V，最大量化误差是多少？计算理论上的最大 SQNR，用分贝表示。

2.19 峰峰值为 5 V 的正弦信号用 16 位 ADC 进行量化，假定采用线性量化，求

(1) 量化步长；

(2) 信号量化噪声比的均方根。

阐述任何所做的假设。

2.20 DSP 系统的模拟输入以 100 kHz 的速率被数字化，采用均匀量化。假定正弦波输入的峰峰值为 ±5 V，为了达到至少 90 dB 的 SQNR，求 ADC 最小的位数，阐述任何所做的合理假设。

2.21 证明：线性 ADC 的信号量化噪声比为

$$\text{SQNR} = 6.02B + 4.77 - 20\log(A/\sigma_x)\ (\text{dB})$$

其中 B 是 ADC 的位数，$\pm A$ 是 ADC 的输入范围，σ_x 是输入信号的均方根值，如果 ADC 的分辨率是 16 位，求下面两种输入时的 SQNR：

(1) 正弦波信号；

(2) 均方根为 $A/4$ 的信号。

阐述任何所做的假设。

2.22 输入到 B 位 ADC 的模拟输入信号的均方根为 σ_x(V)，ADC 的输入范围被调整为 $\pm 3\sigma_x$(V)，求转换器的 SQNR 的表达式，用分贝表示。阐述任何所做的合理假设。

A/D 转换的过抽样 – 混叠和量化噪声控制

2.23 (a) 用图解法解释过抽样技术的原理，它们是如何用来提高奈奎斯特率模数转换器的有效分辨率的。

(b) 数字音频系统使用过抽样技术和 8 位双极型的奈奎斯特率模数转换器来数字化模拟输入信号，模拟输入信号的频率范围为 0 ~ 4 kHz。如果抽样率为 40 MHz，估计转换器的有效分辨率（用位表示）。说明你是如何得到答案的，解释与该方法有关的实际问题。

2.24 (a) 为了通用的目的，收集医学数据的多通道（最多64通道）数据采集系统存在一定的要求。每个模拟通道由用户分别构造，通带的带沿频率在0.5 Hz和200 Hz之间，可选择的抽样频率在1 Hz到2 kHz的范围内。在通带内，最大可允许的波纹是0.5 dB，像频分量必须至少低于信号分量40 dB以下。

解释为了满足以上要求必须采用的策略，你的答案应该包括如下几点：

(i) 特定应用问题的考虑；

(ii) 为了满足有效和经济（根据成本/元件数）的要求，在该应用中如何使用过抽样技术。

(b) 假定在(a)中，系统的所有通道采用相同的抗混叠滤波器，每个滤波器具有下列的巴特沃斯特性：

$$A(f)=\frac{1}{\sqrt{1+\left(\frac{f}{f_c}\right)^8}}$$

其中f_c为滤波器的3 dB截止频率。

借助抽样前后数据频谱的图解求

(1) 截止频率f_c；

(2) 一个合适的公共抽样频率F_s。

解释你的结果。

2.25 音频系统处理基带信号，基带频率从0～20 kHz，求过抽样比和为了实现用8位转换器达到用16位转换器得到的性能所必需的最小抽样频率。

2.26 (a) 结合一位ADC，简要说明下列技术：

(i) 过抽样；

(ii) 噪声谱整形。

(b) 为什么一位ADC更适合于高保真DSP系统中常规的渐次逼近？

(c) 一个具有模拟音频信号输入的数字信号处理系统，输入信号的频率范围为0～20 kHz，采用过抽样技术和图2.53所描述的一阶Σ-Δ调制器来数字化模拟信号。假定抽样频率为3.072 MHz，求噪声整形滤波器在20 kHz处的频率响应。估计数字化转换器的有效分辨率，用位表示。

2.27 一个具有模拟音频信号输入的数字信号处理系统，输入信号的频率范围为0～20 kHz，采用过抽样技术和二阶Σ-Δ调制器将模拟信号转换成速率为6.144 MHz的数字比特流。Σ-Δ调制器的z平面模型如图2.54所示。

求数字转换器通过过抽样和噪声整形的信号量化噪声比的总的改善，以及数字转换器的有效分辨率，用位表示。

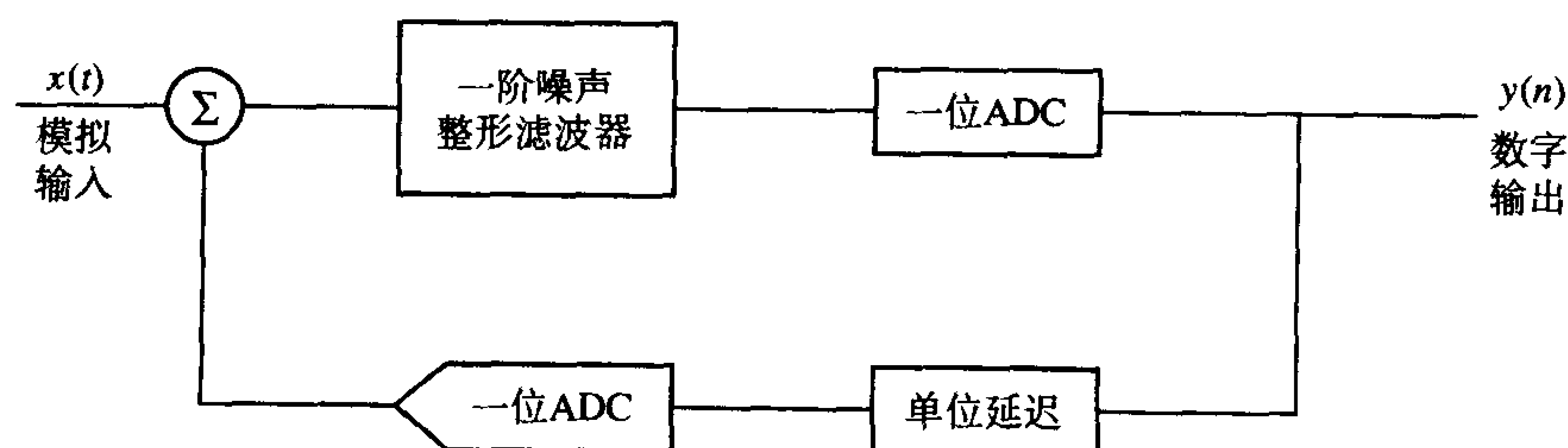


图2.53 一阶Σ-Δ调制器

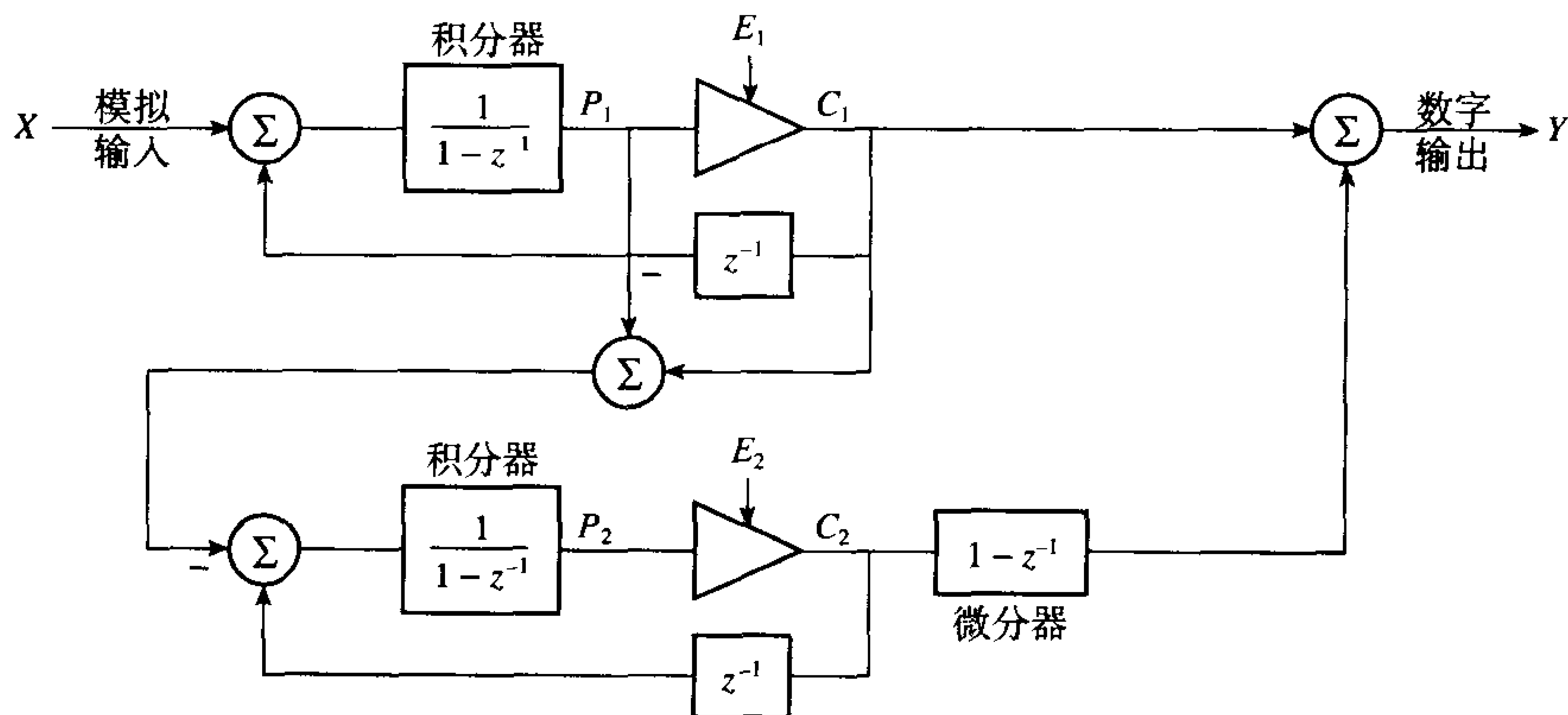


图 2.54 二阶 Σ-Δ 调制器的输出变换 $Y(z) = X(z) + E_2(z)\,(1-z^{-1})^2$

2.28 一个模拟音频信号输入频率范围为 0 ~ 20 kHz 的数字信号处理系统使用过抽样技术和一阶 Σ-Δ 调制器，将模拟信号转换成速率为 6.144 MHz 的数字比特流。Σ-Δ 调制器的 z 域模型如图 2.55 所示。

(i) 解释数字比特流是如何转换成速率为 92 kHz 的数字多比特流（multibit stream）的；

(ii) 求用过抽样和噪声整形对信号量化噪声比的可能的总改善，即求数字化的有效分辨率，用位表示。

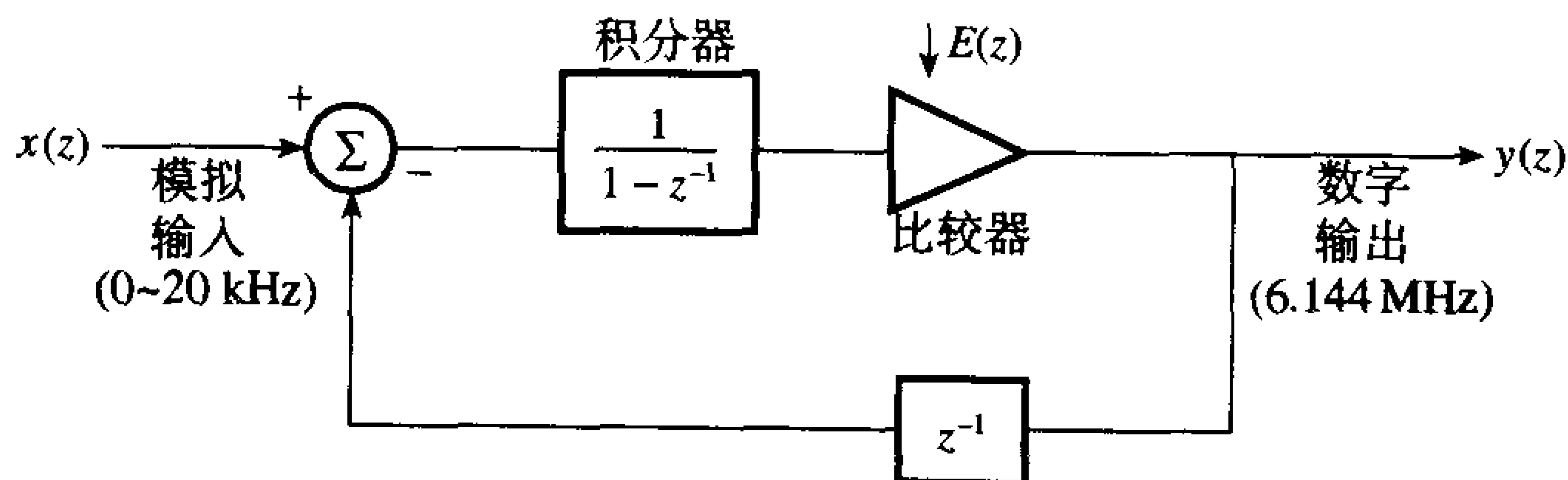
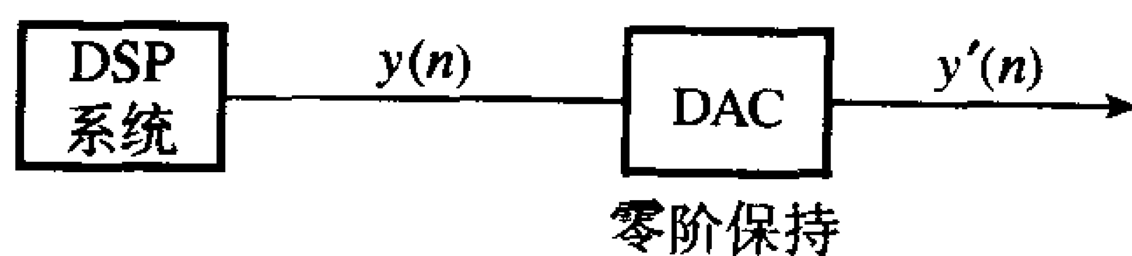


图 2.55 SDM z 域模型

D/A 转换和 sin x/x 效应

2.29 输入为模拟信号的 DSP 系统如图 2.56(a)所示，图 2.56(b)画出了加到 DAC 的信号的基带频谱。画出 DAC 的输出端信号在 0 ~ $2F_s$ 间隔上的频谱，其中 F_s 是抽样频率，在你的图中求信号分量的幅度。假定抽样频率为 15 kHz。



(a) 具有模拟输出的DSP系统



(b) 加到DAC的信号的频谱

图 2.56 具有模拟输出的 DSP 系统和加到 DAC 的信号的频谱

2.30 某实时 DSP 系统使用 12 位处理器、转换时间为 15 μs 的 6 位 ADC 以及建立时间为 500 ns 的 12 位 DAC，如果要求的 DSP 运算是由下式给出的卷积和：

$$y(n)=\sum_{k=0}^{N-1}h(k)x(n-k)$$

其中变量是通常的含义，计算必须在样本两个抽样值之间执行，估计系统的实时能力，阐述所做的任何假设。

2.31 响应数字序列的数模转换器的输出由下式给出：

$$y(t)=\sum_{n}y(n)h(t-nT)$$

其中 $h(t)$是 DAC 的冲激响应，$1/T$ 是数据输入到 DAC 的速率。假定 DAC 是零阶保持，$h(t)$是持续时间为 T(s)的方波脉冲。

画出 DAC 响应于输入序列 $y(n)$的输出，输入序列 $y(n)$如图 2.57 所示。证明：DAC 对信号频谱的影响可以通过具有如下频率特性的数字滤波器进行补偿：

$$|H(\omega)|=\frac{\omega T}{2\sin(\omega T/2)}$$

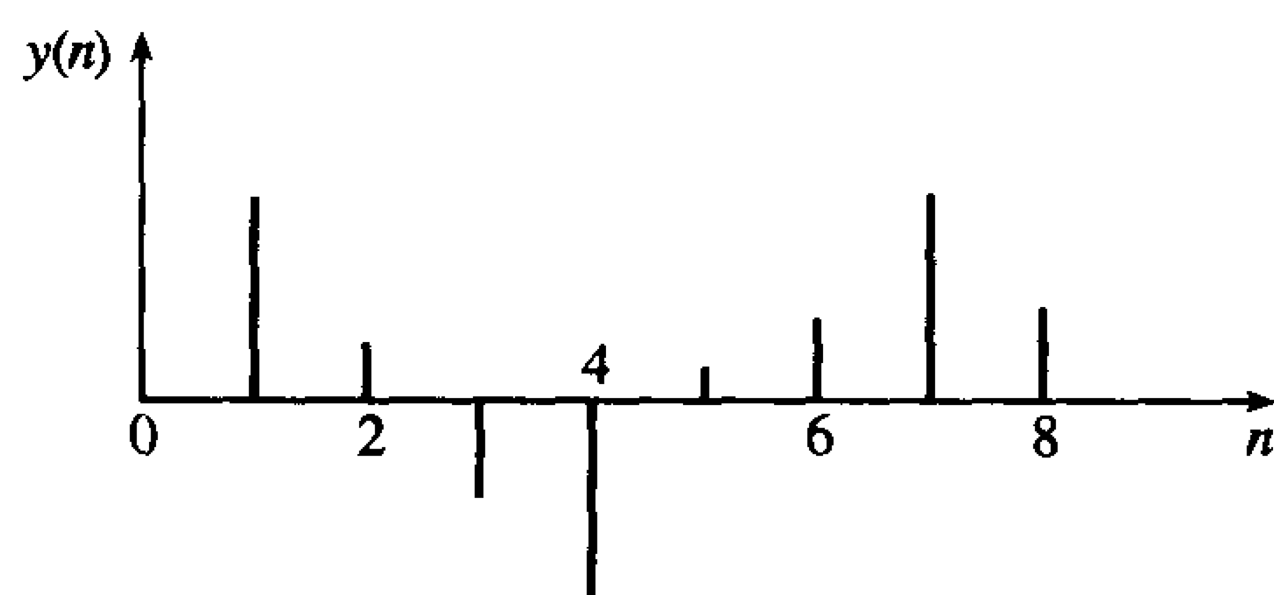


图 2.57 输入序列 $y(n)$

2.32 仔细考察在实时数字信号处理中的约束和由模数转换过程引入的误差。对如何减少这些约束或误差提出建议。

D/A 转换中的过抽样－像频和量化噪声控制

2.33 图 2.58 描述了某实时数字音频系统中信号经数字处理后恢复模拟信号的装置，模拟信号的基带频率范围从 0 到 24 kHz，抽样率是 192 kHz。

像频至少要被抑制 50 dB，而音频信号分量的变化不超过 0.5 dB，借助信号频谱图求抗镜像滤波器的阶数和截止频率。假定抗镜像滤波器具有巴特沃斯特性，阐述所做的任何合理的假设。

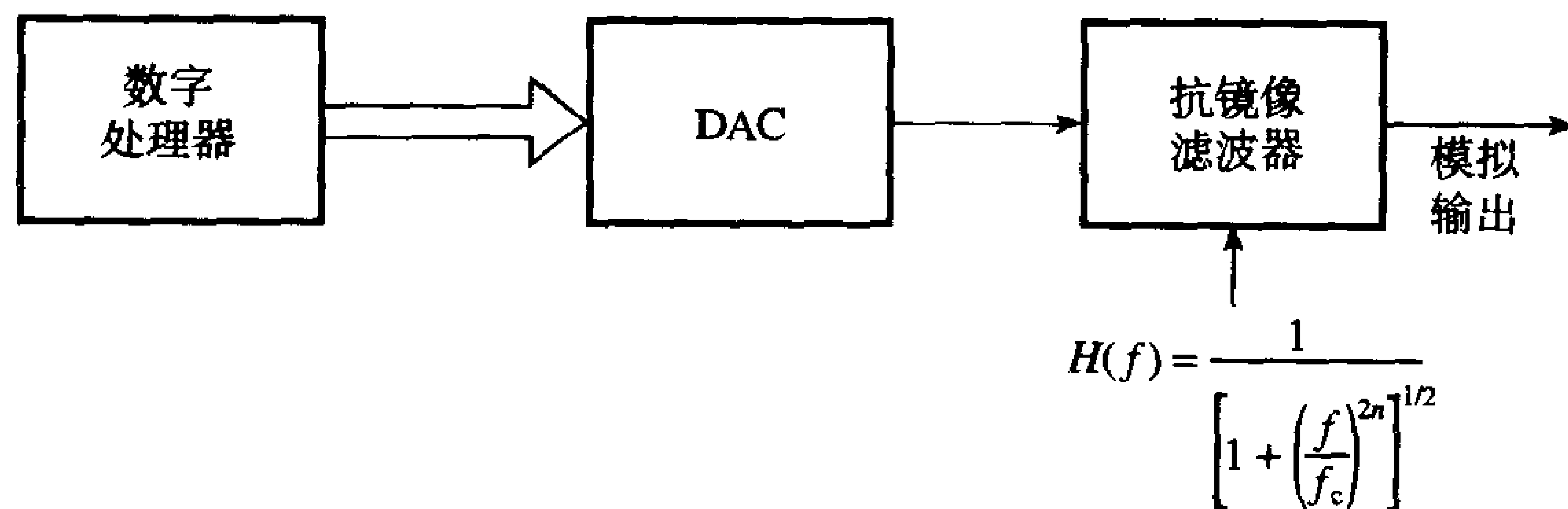


图 2.58 某实时 DSP 系统的后端

2.34 图 2.59 描述了某实时数字音频系统中信号经数字处理后恢复模拟信号的装置，模拟信号的基带频率范围从 0 到 24 kHz，抽样率是 176.4 kHz。

噪声整形器由下列方程来刻画：

$$y'(n)=x(n)-e(n-1) \tag{a}$$

$$e(n)=y(n)-y'(n) \tag{b}$$

(a) 推导由量化噪声看到的传递函数表达式，并画出噪声整形后量化噪声的频谱。

(b) 求由过抽样和噪声整形可能带来的信号量化噪声比的改善，即估算 DAC 的有效分辨率，用位数表示。假定采用四倍过抽样比。

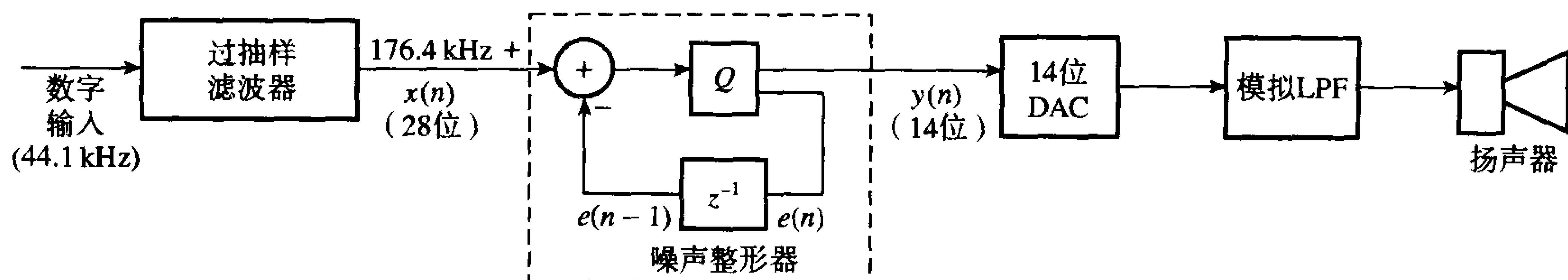


图 2.59　CD 播放器中用四倍过抽样和噪声整形的音频信号重现的简化框图

2.35 某 DSP 系统前接抽样保持和孔径时间为 10 ns、采集时间为 1 μs 的 8 位 ADC，求最大支持 100 kHz 的抽样频率的最大 ADC 转换时间。

MATLAB 习题

2.36 图 2.59 描述了某实时数字音频系统中信号经数字处理后恢复模拟信号的装置，模拟信号的基带频率范围从 0 ~ 20 kHz，抽样率是 176.4 kHz。

(a) 用 MATLAB 计算并画出四倍过抽样滤波器的频谱，列出滤波器的系数；

(b) 用 12 位定点数表示滤波器系数；

(c) 产生一个音频信号（在 44.1 kHz 处，用 16 位表示）；

(d) 在 MATLAB 中用定点算术模拟 D/A 过程，画出时域和频域每块输出端的信号。

2.37 带宽为 B、载波频率为 f_c 的通信信号的频谱如图 2.60(a)所示，模拟信号通过抗混叠滤波器，以 F_s 的频率进行抽样，希望的抽样后信号频谱如图 2.60(b)所示。

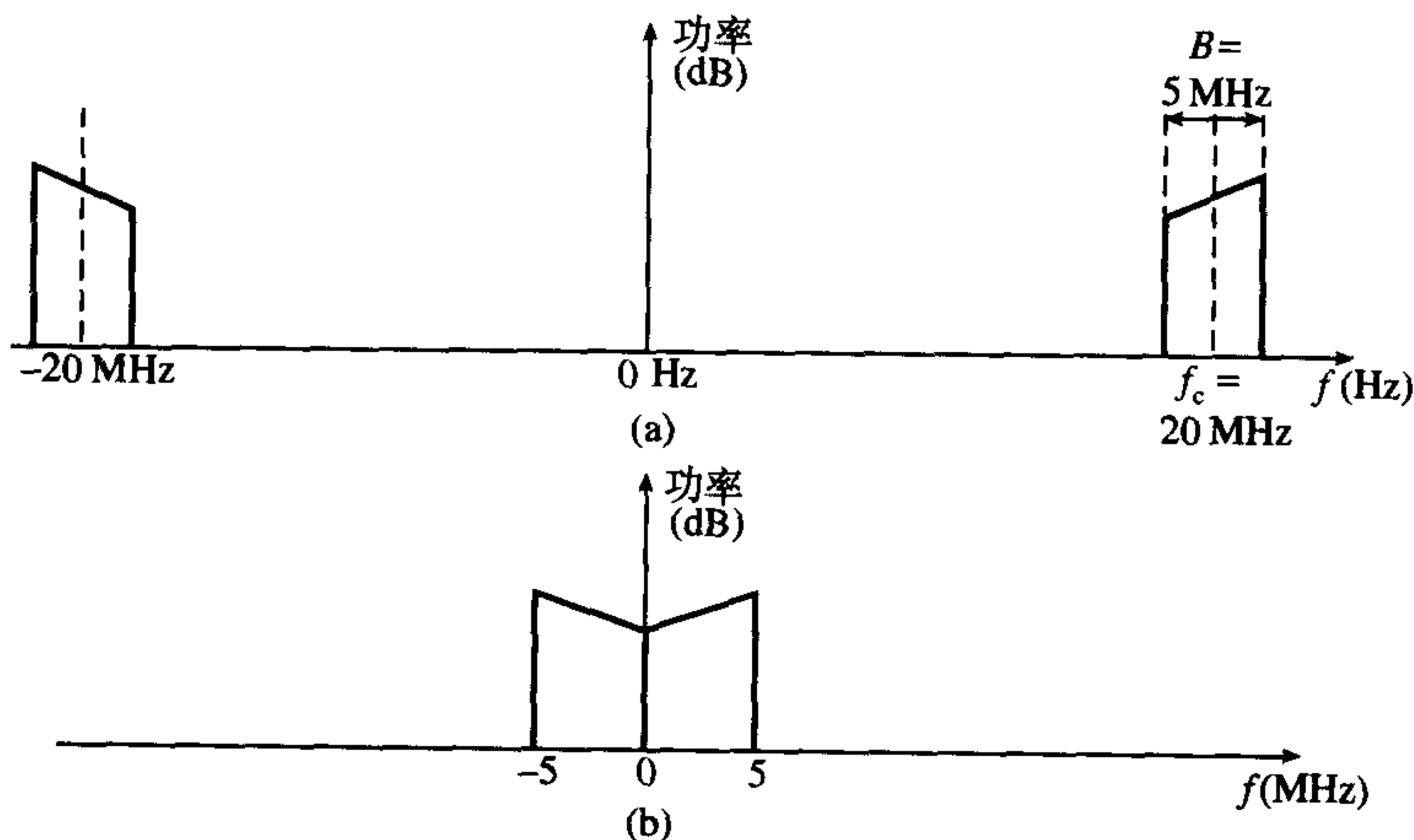


图 2.60　习题 2.37 的图示

(1) 提出抗混叠滤波器的技术规范并加以论证；

(2) 提出一组合适的抽样频率，使图 2.60(b)的频谱能够恢复而不产生混叠，对每种情况画出得到的频谱。

(3) 为了使抽样率尽可能地保持很低，选择(2)中的最低抽样频率，并且用 MATLAB 验证次抽样后的信号（sub-sampled signal）的频谱，在你的答案中提供 MATLAB 代码。

提示： 抽样在时域是相乘，在数学上等价于频域的卷积。检查抽样函数的频谱，并将它与图 2.60(a)中的频谱卷积。

(4) 提出恢复感兴趣信号的数字滤波器的技术规范，讨论加上防护频带是怎样有助于简化这一滤波器的。

2.38 (a) 在框图级设计一个简单的数字AM接收机，接收机使用带通过抽样技术和正交混频来解调接收的信号。假定IF带宽是6 kHz。你的设计应该包括：

- 技术规范（加以说明）：合适的IF中心频率，范围在40 ~ 60 kHz，避免混叠的最佳抽样频率，合适的正交振荡器频率，以及合适的数字滤波器；
- 画出抽样前后IF信号的频谱；
- 描述数字接收机是如何工作的；
- 阐述任何合理的假设。

(b) 建立并且测试数字AM接收机的简化的MATLAB模型。

参考文献

CCITT (1989) Possible applications for 16 kbits/sec voice coding. Appendix 3 – Annex 1 to Question 21/XV, 13–22, March.

CCITT Recommendation G.726 (1990) 40, 32, 24 and 16 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM), ITU Geneva, Switzerland.

CCITT Recommendation G.711 (1998) Pulse Code Modulation (PCM) of Voice Frequencies, ITU Geneva, Switzerland.

Del Re E. (1978) Bandpass signal filtering and reconstruction through minimum-sampling-rate digital processing. *Alta Frequenza*, **47**(9), September, 395E/675–398E/678.

Goedhart D., Van de Plassche R.J. and Stikvoort E.F. (1982) Digital-to-analog conversion in playing compact disc. *Philips Technical Rev.*, **40**(6), 174–9.

Vaughan R.G., Scott N.L. and White D.R. (1991) The theory of bandpass sampling. *IEEE Transactions on Signal Processing*, **39**(9), September, 1973–84.

参考书目

Aziz P.M., Sorensen H.V. and Spiegel J.V.D. (1996) An overview of sigma-delta converters. *IEEE Signal Processing Magazine*, January, 61–84.

Bellamy J. (1982) *Digital Telephony*. New York: John Wiley & Sons.

Berkhout P.J. and Eggermont L.D.J. (1985) Digital audio systems. *IEEE ASSP Magazine*, October, 45–67.

Betts J.A. (1978) *Signal Processing, Modulation and Noise*. Unibooks, Hodder and Stoughton.

Blesser B.A. (1978) Digitization of audio: a comprehensive examination of theory, implementation, and current practice. *J. Audio Eng. Soc.*, **26**(10), 739–71.

Blesser B., Locanthi B. and Stockham Jr, T.G. (eds) (1982) *Digital Audio*. New York: Audio Engineering Society.

Candy J.C., Wooley B.A. and Benjamin O.J. (1981) A voice band codec with digital filtering. *IEEE Trans. Communications*, **COM-29**(6), June, 815–30.

Garret P.H. (1981) *Analog I/O Design*. Reston VA: Reston Publishing Co. Inc.

ITTCC (1986) Study Group XVIII – Report R26C, Recommendation G7221. 32 kbit/s Adaptive Differential Pulse-Code Modulation (ADPCM).

Jayant N.S. and Noll P. (1984) *Digital Coding of Waveforms*. Englewood Cliffs NJ: Prentice-Hall.

Macario R.C.V. (1991) *Signal Coding B: Speech Coding*. C. Xydeas, 82–99.

Mueller H.R., Schindler H.R. and Vettiger P. (1978) Signal-to-noise analysis of a PCM voice system ba analogue/digital filtering. *IEEE Trans. Communications*, **COM-26**(5), May, 653–9.

Natvig J.E. (1988) Speech coding in the pan-European digital mobile radio systems. *Speech Communication Magazine*, January.

Oliver B.M., Pierce J.R. and Shannon C.E. (1948) The philosophy of PCM. *Proc IRE*, November, 1324–31.

Oppenheim A. and Schaffer R.W. (1975) *Digital Signal Processing*. Englewood Cliffs NJ: Prentice-Hall.

Papamichalis P. (1987) *Practical Approaches to Speech Coding*. Englewood Cliffs NJ: Prentice-Hall.

Rabiner L.R. and Gold B. (1975) *Theory and Applications of Digital Signal Processing*. Englewood Cliffs NJ: Prentice-Hall.

Sheingold D.H. (ed.) (1986) *Analog-Digital Conversion Handbook*. Englewood Cliffs NJ: Prentice-Hall.

Steer Jr, R.W. (1989) Antialiasing filters reduce errors in A/D converters. *EDN*, March, 171–86.

Tiefenthaler C. (1987) Oversampling to increase signal to noise ratio of ADCs. *Electronic Product Design*, March, 59–62.

Van Doren A.H. (1982) *Data Acquisition Systems*. Reston VA: Reston Publishing Co. Inc.

第3章 离散变换

本章包括了在数字信号处理中离散变换应用的入门性的内容，以及广泛应用的快速傅里叶变换的推导。离散余弦、沃尔什（Walsh）、哈达玛（Hadamard）变换也在本章进行了简单的讨论。本章还讨论了小波变换，因为人们对它的兴趣日益增加；同时解释了小波分析，以及基于多分辨分析的信号去噪和奇异检测的应用。

3.1 引言

本章描述了离散数据在时域和频域之间进行的变换，电压与时间的表示变成了幅度和频率以及相位与频率的表示；反之亦然。时域和频域对同样的数据提供互补的信息，因此，在应用中考察在某个特定的频率电压幅度变化的幅度频率图，要比按顺序观察电压波形可能更有意义。例如，通过对输出数据进行快速傅里叶变换，可以得到机器磨损的早期指示。另一个应用例子是，将离散傅里叶分析仪和示波器应用到通信系统的调制器的输出检查中，以确保正确地运行。在这种情况下，测试信号应该在某些已知的频率出现幅度分量。注意到这两种谱分析的情况是选择或者限制一组频率，这说明变换可以利用数据而忽略无关紧要的数据，使得更容易解释数据。离散变换特别是离散余弦变换用在语音和视频信号的数据压缩中，使得这些信号能够以小的带宽进行传输。离散变换也应用于图像处理中，从而为模式识别得到简化的特征集。在其他的信号处理应用中，如在声呐的距离检测中用到的相关、确定系统之间内部关系和输入输出之间关系的卷积和反卷积等，变换是加快计算的有用的数学工具。对于这些计算，从频域到时域的变换如同从时域到频域的变换一样重要。讨论的整个内容是非常数学化的，但是在许多应用中的离散变换现在已经标准化了，所以对于应用工程师来说并不要求掌握数学和相关理论的知识。波形的频谱分析是个例外，这里的每一个问题都必须根据它自身的准则来进行处理。正确理解这些问题，对于避免与需要获得足够的离散规则数据样本有关的许多缺陷，以及避免混叠、栅栏、谱的泄漏等都是很重要的。这些主题将在第11章详细讨论。

在这些有效的变换中，离散傅里叶变换（DFT）以及它的快速计算算法——快速傅里叶变换（FFT）是最知名的，大概也是最重要的。其理由是他们允许在频域表示所有的信号，哪怕是最短的数据记录长度（<1 s），截断的傅里叶频率分量要比任何其他指数型级数能更好地表示数据，每个分量是正弦的，在通过线性系统时不会产生失真，因而构造了一个好的测试信号，并且可以快速计算FFT。另一个理由是傅里叶分析自1822年由傅里叶发表以来就已经存在，因此傅里叶变换得到了高度的重视和开发，因而应用领域十分广泛。

近年来，人们在小波变换方面进行了相当大的努力，因为它可以根据小波幅度描述随机信号的时频变化内容。这个主题高度地数学化，本书通过给出噪声中提取信号的两个说明性例子来讲解其基本原理。

电气和电子工程专业的学生最初学习用拉普拉斯（Laplace）变换来分析电路的电气行为，这是因为傅里叶变换既不能处理非零条件，也不能处理阶跃输入。当他们进一步研究离散系统（如有限冲激响应滤波器）的频率响应和稳定性的时候，则要用到z变换。因此，傅里叶变换的应用主要

出现在使用FFT的快速信号处理计算中和谱分析中。然而，三种变换是有联系的，可以认为拉普拉斯变换更具有一般性，因为其他两种变换是从它推导出来的。因此，拉普拉斯变量是$s=\sigma+\mathrm{j}\omega$，而傅里叶变换变量是$s=\mathrm{j}\omega$，$z$变换变量由$z=\mathrm{e}^{sT}$给出，其中$T$是离散样本值之间的时间。最后，傅里叶变换与$z$变换通过$z=\mathrm{e}^{\mathrm{j}\omega T}$建立起联系（参见第4章）。

3.1.1 傅里叶级数

任何周期波形$f(t)$可以表示为无限正弦项、余弦项和一个常数项之和，这种表示是由下式给出的傅里叶级数，

$$f(t)=a_0+\sum_{n=1}^{\infty}a_n\cos(n\omega t)+\sum_{n=1}^{\infty}b_n\sin(n\omega t) \tag{3.1}$$

其中t是一个独立的变量，它常常表示时间，但是也可以表示距离或任何其他的量；$f(t)$常常表示电压与时间的变化波形，但也可能是其他波形；$\omega=2\pi/T_\mathrm{p}$称为一次谐波、基波或角频率，它与基频f通过$\omega=2\pi f$建立起联系；T_p是波形的重复周期，

$$a_0=\frac{1}{T_\mathrm{p}}\int_{-T_\mathrm{p}/2}^{T_\mathrm{p}/2}f(t)\,\mathrm{d}t$$

是常数，它等于$f(t)$在一个周期上的时间平均，它可能代表dc电压，

$$a_n=\frac{2}{T_\mathrm{p}}\int_{-T_\mathrm{p}/2}^{T_\mathrm{p}/2}f(t)\cos(n\omega t)\,\mathrm{d}t$$

和

$$b_n=\frac{2}{T_\mathrm{p}}\int_{-T_\mathrm{p}/2}^{T_\mathrm{p}/2}f(t)\sin(n\omega t)\,\mathrm{d}t$$

频率$n\omega$称为ω的n次谐波，因此，无穷级数3.1式包括频率相关的正弦项和余弦项，这些正弦项和余弦项在正的谐波频率$n\omega$处的幅度a_n和b_n是不同的。这种级数用指数形式表示可以写得更为紧凑一些，指数形式表示的优点是更易于数学上的处理。那么，级数就变成了如下形式：

$$f(t)=\sum_{n=-\infty}^{\infty}d_n\,\mathrm{e}^{\mathrm{j}n\omega t} \tag{3.2}$$

其中

$$d_n=\frac{1}{T_\mathrm{p}}\int_{-T_\mathrm{p}/2}^{T_\mathrm{p}/2}f(t)\mathrm{e}^{-\mathrm{j}n\omega t}\,\mathrm{d}t \tag{3.3}$$

是复数，$|d_n|$具有伏特单位。

求和包括n的负值，所以一半级数是由负频率$-n\omega$组成的。负频没有物理意义，纯粹是数学上的一种表示方法。于是，复幅度d_n的大小$|d_n|$在数值上二等分，这表示对应的正频和负频之间幅度的均等。因此，在频率$n\omega$的正确幅度由计算出的值乘以2得到。复幅度和三角形式的幅度具有如下关系：

$$|d_n|=(a_n^2+b_n^2)^{1/2} \tag{3.4}$$

和

$$\phi_n = -\tan^{-1}(b_n/a_n) \tag{3.5}$$

其中ϕ_n是第n次谐波分量的相角，由d_n的虚部和实部之比的反正切给出。因此，波形的每个谐波分量由它的相角和幅度来刻画。

例 3.1　周期性单极脉冲如图3.1(a)所示，有意选择时间的起点偏移脉冲的中心和边沿，这是为了说明傅里叶级数的相位特征。在3.3式中代入适当的值，得

$$\begin{aligned}
d_n &= \frac{1}{T_p}\int_{-(\tau-x\tau)}^{x\tau} A\,\mathrm{e}^{-\mathrm{j}n\omega t}\,\mathrm{d}t \\
&= \frac{A}{T_p}\left[\frac{\mathrm{e}^{-\mathrm{j}n\omega t}}{-\mathrm{j}n\omega}\right]_{-(\tau-x\tau)}^{x\tau} \\
&= \frac{A}{n\omega T_p}\,\frac{\mathrm{e}^{-\mathrm{j}n\omega x\tau}-\mathrm{e}^{\mathrm{j}n\omega(\tau-x\tau)}}{-\mathrm{j}} \qquad (3.6) \\
&= \frac{A}{n\omega T_p}\mathrm{e}^{-\mathrm{j}n\omega x\tau}\left[\frac{\mathrm{e}^{\mathrm{j}n\omega\tau}-1}{\mathrm{j}}\right] \\
&= \frac{2A}{n\omega T_p}\mathrm{e}^{-\mathrm{j}n\omega x\tau}\left[\frac{\mathrm{e}^{\mathrm{j}n\omega\tau/2}-\mathrm{e}^{-\mathrm{j}n\omega\tau/2}}{2\mathrm{j}}\right]\mathrm{e}^{\mathrm{j}n\omega\tau/2} \\
&= \frac{2A}{n\omega T_p}\mathrm{e}^{\mathrm{j}n\omega(\tau/2-x\tau)}\sin\left(\frac{n\omega\tau}{2}\right) \\
&= \frac{2A}{n\omega T_p}\,\frac{n\omega\tau}{2}\,\mathrm{e}^{\mathrm{j}n\omega(\tau/2-x\tau)}\,\frac{\sin(n\omega\tau/2)}{n\omega\tau/2} \qquad (3.7) \\
&= \frac{A\tau}{T_p}\mathrm{Sa}\left(\frac{n\omega\tau}{2}\right)\mathrm{e}^{\mathrm{j}n\omega(0.5-x)\tau}
\end{aligned}$$

其中

$$\mathrm{Sa}\left(\frac{n\omega\tau}{2}\right) = \frac{\sin(n\omega\tau/2)}{n\omega\tau/2}$$

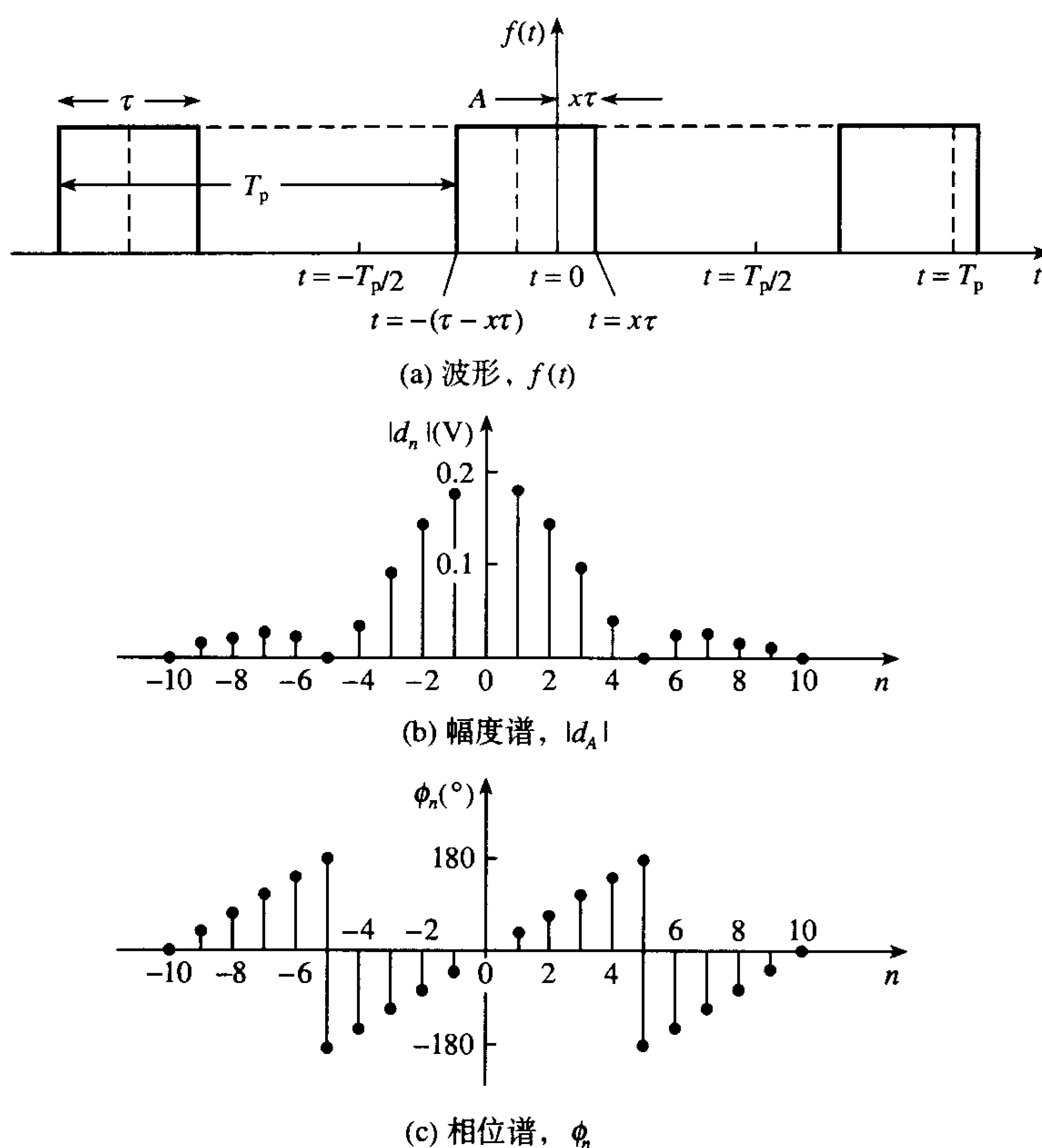
称做自变量为$n\omega\tau/2$的抽样函数。d_n的模为

$$|d_n| = \frac{A\tau}{T_p}\left|\mathrm{Sa}\left(\frac{n\omega\tau}{2}\right)\right|$$

图3.1(b)画出了模。$n\omega(0.5-x)\tau$表示与第n次谐波分量有关的相角ϕ_n，相角用弧度表示。为了能够画出相角与谐波数n的关系，考虑一种特殊的情况。令$x=0$，即定位在时间的原点在脉冲的后沿。令$\tau=T_p/5$，当

$$\phi_n = \frac{n\omega\tau}{2} = n\frac{2\pi}{T_p}\frac{\tau}{2} = n\frac{2\pi}{T_p}\frac{T_p}{5}\frac{1}{2} = \frac{n}{5}\pi$$

ϕ_n画在图3.1(c)中，其中约定$-180° \leqslant \phi_n \leqslant 180°$。不同时间原点的选择可能导出不同的相位谱（$\phi_n$与$n$的关系），在对称位置定位时间的原点通常使分析简化。例如，在周期脉冲链中，时间原点定位在脉冲的中点。对于所选的情况可以看出，幅度谱（参见图3.1(b)）是偶函数（$|d_n|=|d_{-n}|$），而相位谱（参见图3.1(c)）是奇函数（$\phi_{-n}=-\phi_n$），相角ϕ_n、ϕ_{-n}给出了谐波分量相对于另一个谐波分量的相对相角。在时间t，由3.2式绝对相角为$\{n\omega(0.5-x)\tau+n\omega t\}$。



(a) 波形，$f(t)$

(b) 幅度谱，$|d_A|$

(c) 相位谱，ϕ_n

图 3.1　周期性单极脉冲的波形、幅度谱和相位谱

3.1.2　傅里叶变换

傅里叶级数方法当波形是非周期的时候必须加以修正，一个很重要的例子是从图3.1(a)中当周期 T_p 趋于无穷时得到的单个矩形脉冲的情况。当 T_p 增加时，谐波分量之间的间隔 $1/T_p = \omega/2\pi$ 减小到 $\mathrm{d}\omega/2\pi$，最终变成零，这对应于从离散频率变量 $n\omega$ 变到连续变量 ω，幅度谱和相位谱变成连续的。因此，当 $T_p \to \infty$ 时，$d_n \to d(\omega)$，通过这些修改，3.3 式变成

$$d(\omega) = \frac{\mathrm{d}\omega}{2\pi}\int_{-\infty}^{\infty} f(t)\,\mathrm{e}^{-\mathrm{j}\omega t}\,\mathrm{d}t \tag{3.8}$$

上式通常除以 $\mathrm{d}\omega/2\pi$ 来归一化，得

$$\frac{d(\omega)}{\mathrm{d}\omega/2\pi} = F(\mathrm{j}\omega) = \int_{-\infty}^{\infty} f(t)\,\mathrm{e}^{-\mathrm{j}\omega t}\,\mathrm{d}t \tag{3.9}$$

$F(\mathrm{j}\omega)$是复数，称为傅里叶积分，或者更为常用的称为傅里叶变换。如果我们表示 $F(\mathrm{j}\omega)$为

$$F(\mathrm{j}\omega) = \mathrm{Re}\,(\mathrm{j}\omega) + \mathrm{j}\,\mathrm{Im}\,(\mathrm{j}\omega) = |F(\mathrm{j}\omega)|\,\mathrm{e}^{\mathrm{j}\phi(\omega)} \tag{3.10}$$

那么

$$|F(\mathrm{j}\omega)| = [\mathrm{Re}^2\,(\mathrm{j}\omega) + \mathrm{Im}^2\,(\mathrm{j}\omega)]^{1/2} \tag{3.11}$$

单位为每赫兹伏特而不是伏特，因此 $|F(\mathrm{j}\omega)|$ 是幅度密度，称为幅度频谱密度。相位角 $\phi(\omega)$为

$$\phi(\omega) = \tan^{-1}[\mathrm{Im}(\mathrm{j}\omega)/\mathrm{Re}(\mathrm{j}\omega)] \tag{3.12}$$

$|F(\mathrm{j}\omega)|^2$ 的单位为 $\mathrm{V^2 Hz^{-2}}$。由于归一化了功率，所以由 1 Ω 电阻消耗的功率单位为 $\mathrm{V^2}$，等价于 $\mathrm{J\,s^{-1}}$，或者 J Hz（J 表示焦耳，能量的单位），那么 $\mathrm{V^2 Hz^{-2}}$ 是 $\mathrm{J\,Hz \times Hz^{-2} = J\,Hz^{-1}}$。因此 $|F(\mathrm{j}\omega)|^2$ 的单位等价于能量 $\mathrm{Hz^{-1}}$，即 $|F(\mathrm{j}\omega)|^2$ 是能量谱密度。$|F(\mathrm{j}\omega)|$ 与 f 的图中在频率 $f_0-\mathrm{d}f$ 与 $f_0+\mathrm{d}f$ 之间的面积给出了频率 f_0 处的平均电压；同样，$|F(\mathrm{j}\omega)|^2$ 与 f 的图中的对应面积给出了频率 f_0 处的平均能量。频谱密度与频率的关系在谱分析中是相当常见的。

例 3.2 回到前面单个脉冲情况的讨论，我们现在用 3.9 式和图 3.1(a)计算幅度谱密度，表达式变成

$$F(\mathrm{j}\omega) = \int_{-(\tau-x\tau)}^{x\tau} A\,\mathrm{e}^{-\mathrm{j}\omega t}\,\mathrm{d}t \tag{3.13}$$

上式与 3.6 式的不同之处是相差一个常数 $1/T_p$，我们得到

$$F(\mathrm{j}\omega) = A\tau\,\mathrm{e}^{\mathrm{j}\omega(1/2-x)\tau}\,\mathrm{Sa}(\omega\tau/2) \tag{3.14}$$

它比 d_n 大一个比例因子 T_p，对应于这样一个事实：$|F(\mathrm{j}\omega)|$ 的单位为伏特乘以时间，或者 $\mathrm{V\,Hz^{-1}}$。如果利用傅里叶变换的性质，得到 3.14 式的结果要比得到 3.7 式的更为简单。因此宽度为 τ、中心在 $t=0$ 的单位幅度脉冲表示为 $\mathrm{rect}(t/\tau)$，它的傅里叶变换为 $\tau\mathrm{Sa}(\omega\tau/2)$。由于 $Af(t)$ 的傅里叶变换为 $AF[f(t)]$，其中 F 表示傅里叶变换，那么高度为 A 的脉冲其傅里叶变换为 $A\tau\,\mathrm{Sa}(\omega\tau/2)$。在图 3.1(a)的情况中，脉冲被左移 $\tau/2-x\tau$，实际的矩形脉冲为 $\mathrm{rect}\{[t+(\tau/2-x\tau)]/\tau\}$。傅里叶变换的延迟特性指出，对于右移 t_0 的脉冲，$F[f(t-t_0)] = \mathrm{e}^{-\mathrm{j}\omega t_0}F[f(t)]$。将这一性质应用到 $A\tau\,\mathrm{Sa}(\omega\tau/2)$ 得到要求的变换形式为

$$F(\mathrm{j}\omega) = \mathrm{e}^{+\mathrm{j}\omega(+\tau/2-x\tau)}A\tau\,\mathrm{Sa}\left(\frac{\omega\tau}{2}\right)$$

$$= A\tau\,\mathrm{e}^{\mathrm{j}\omega(1/2-x)\tau}\,\mathrm{Sa}\left(\frac{\omega\tau}{2}\right)$$

上式与 3.14 式相同。

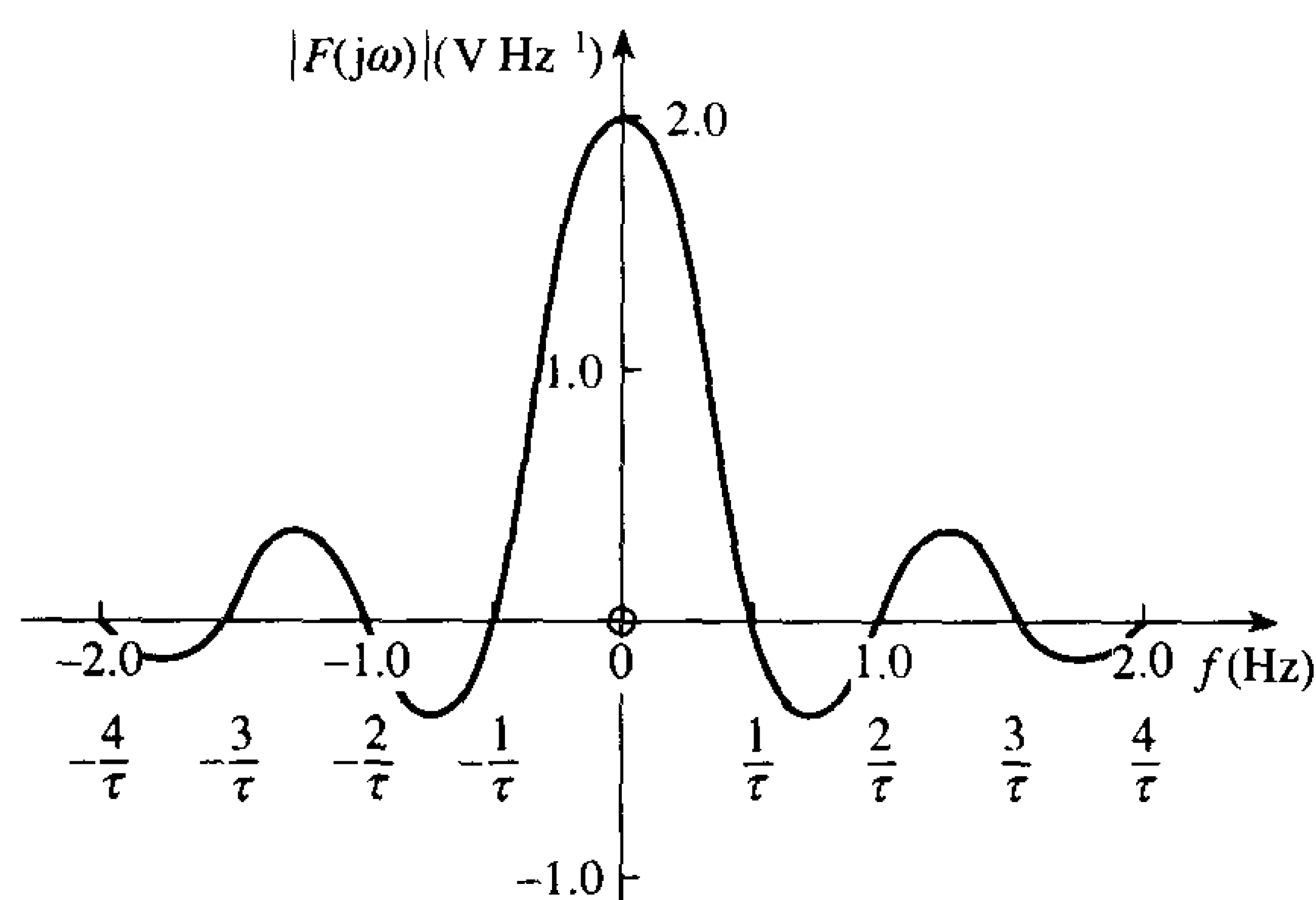
如果时间的原点位于脉冲的中央，即 $x=\frac{1}{2}$，那么，脉冲的傅里叶变换为

$$F(\mathrm{j}\omega) = \frac{A\tau\sin(\omega\tau/2)}{\omega\tau/2} = A\tau\,\mathrm{Sa}\left(\frac{\omega\tau}{2}\right) \tag{3.15}$$

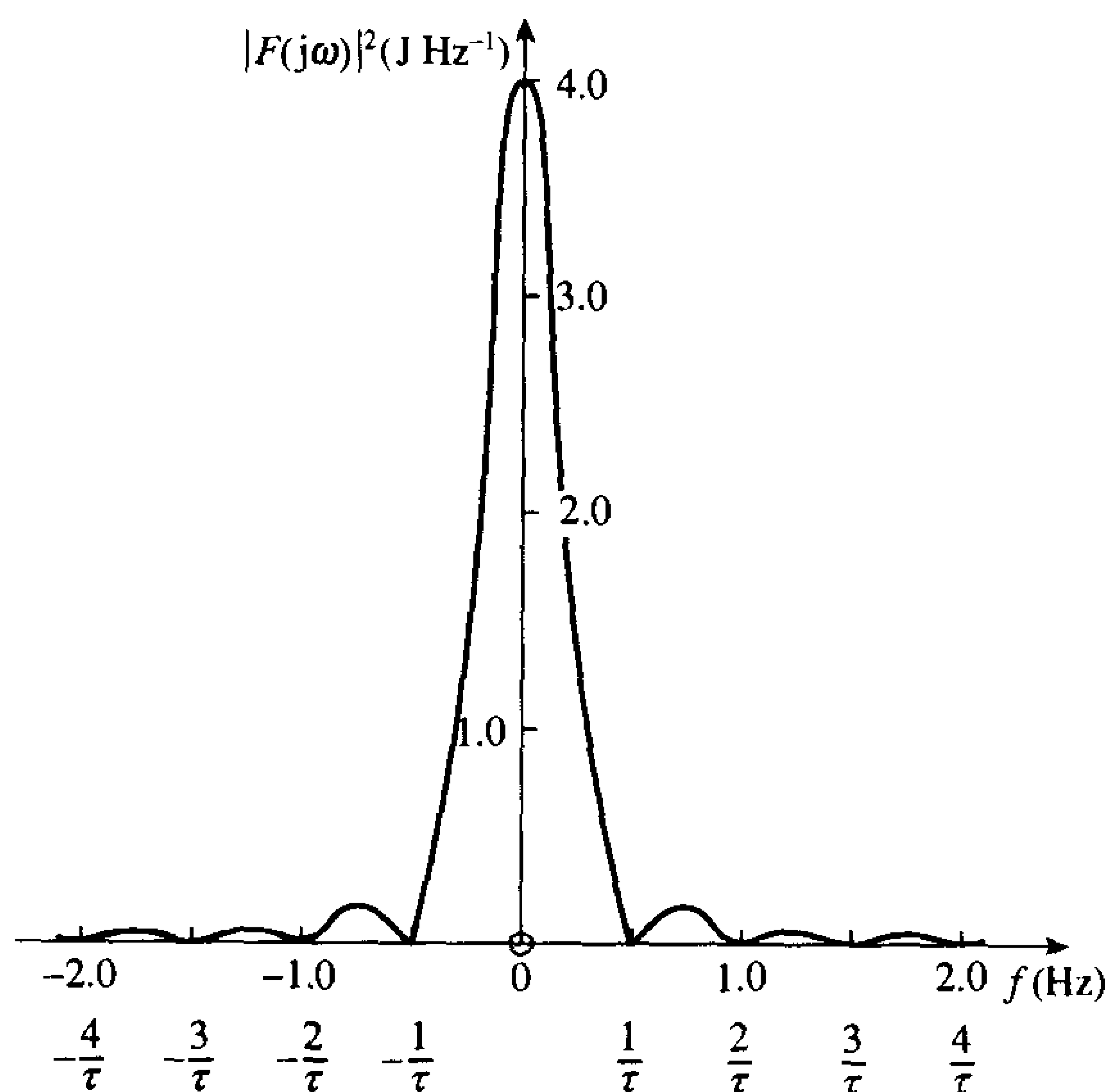
并且傅里叶变换是实的。$|F(\mathrm{j}\omega)|$ 是连续的，对于 $A=1$ V，$T_p=10$ s 和 $\tau=2$ s 的 $|F(\mathrm{j}\omega)|$ 的图形由图 3.2(a)给出。形状正比于抽样函数的幅度谱总是与矩形脉冲以及波形有限持续时间 τ 有关，后者可以看作为无限的波形乘以 $\mathrm{rect}[(t\pm t_0)/\tau]$，即乘以一个单位脉冲。用实验方法确定的波形就属于这一范畴，具有有限持续时间 τ。当 $\sin(\omega\tau/2)=0$ 时抽样函数通过零点，即 $\omega\tau/2=m\pi$（$m\neq 0$，m 是整数）时。因此，在 $f=1/\tau, 2/\tau, 3/\tau, \ldots$ 处幅度为零，当 $\omega\to 0$ 时，$\sin(\omega\tau/2)\to\omega\tau/2$ 和 $\mathrm{Sa}(\omega\tau/2) = \sin(\omega\tau/2)/(\omega\tau/2)\to 1$，所以当 $\omega=0$（即 $f=0$）时，$F(\mathrm{j}\omega)=A\tau$。幅度为 2 V 的脉冲的能量谱密度如图 3.2(b)所示，比较图 3.2(a)的幅度谱。

利用傅里叶反变换从频域变换到时域也是可能的，这时，

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\mathrm{j}\omega)\,\mathrm{e}^{\mathrm{j}\omega t}\,\mathrm{d}\omega = \int_{-\infty}^{\infty} F(\mathrm{j}\omega)\,\mathrm{e}^{\mathrm{j}\omega t}\,\mathrm{d}f \tag{3.16}$$



(a) 2 V 脉冲的幅度谱



(b) 2 V 脉冲的能量谱

图 3.2　2 V 脉冲的幅度谱和能量谱

3.2　DFT 及其逆

在实际中，数据的傅里叶变换是用数字计算机而不是用模拟处理得到的。由于模拟波形由无穷多的比邻的点组成，表示所有这些值在实际中是不可能的。因此，模拟值必须以均匀的间隔进行抽样，然后将抽样值转换成数字二进制表示。这可以用抽样保持接模数转换器来实现，抽样速率要高到足以正确地表示波形。理论上必需的抽样率是奈奎斯特抽样率，即 $2f_{max}$，其中 f_{max} 是有效幅度信号中最高频率正弦分量的频率。在本章中数据的数字值是可以变换的，而像谱分析所要求的加窗（windowing）将在第 11 章讨论。因此，待变换的数据是离散的，可能也是非周期的，它不能应用傅里叶变换，因为傅里叶变换是针对连续数据的，然而可以对离散数据应用离散傅里叶变换（DCT）。

假定波形以均匀的时间间隔 T 进行抽样，得到 N 个抽样值的抽样序列 $\{x(nT)\} = x(0), x(T),..., x[(N-1)T]$，其中 n 是从 $n = 0$ 到 $n = N-1$ 的抽样数，数据值 $x(nT)$ 在表示一个时间序列如电压波形时是实的。那么，$x(nT)$ 的 DFT 定义为频域中的一个复值序列 $\{X(k\Omega)\} = X(0), X(\Omega),..., X[(N-1)\Omega]$，其中 $\Omega = 2\pi/(N-1)T$ 是一次谐波频率，当 $N \gg 1$ 时，$\Omega \simeq 2\pi/NT$。因此，$X(k\Omega)$ 一般有实部和虚部分量，所以，对于 k 次谐波，

$$X(k) = R(k) + \mathrm{j}I(k) \tag{3.17}$$

且

$$|X(k)| = [R^2(k) + I^2(k)]^{1/2} \tag{3.18}$$

$X(k)$的相角为

$$\phi(k) = \tan^{-1}[I(k)/R(k)] \tag{3.19}$$

其中$X(k)$表示$X(k\Omega)$。将3.17式～3.19式与3.10式～3.12式进行比较可以看出，这些等式类似于傅里叶变换的等式。

注意，N个实数据变换成N个复DFT值（频域）。DFT值$X(k)$为

$$X(k) = F_{\mathrm{D}}[x(nT)] = \sum_{n=0}^{N-1} x(nT)\,\mathrm{e}^{-\mathrm{j}k\Omega nT},\ k = 0, 1, \ldots, N-1 \tag{3.20}$$

其中F_{D}表示离散傅里叶变换。在这个等式中k表示变换分量的谐波数，可以看出等式类似于3.9式的傅里叶变换。在3.9式中，当$T<0$或$t>(N-1)T$时$f(t)=0$，其中令$x(nT)=f(t)$、$k\Omega=\omega$和$nT=t$，这时3.20式与3.9式类似，所以，两个变换具有类似的性质是可以期待的，然而变换并不相同。将这些变量替换代入3.9式，且令$\mathrm{d}t=T$，积分用对谐波频率kf_s求和取代，其中$f_s=1/(N-1)T=2\pi/\Omega$，

$$\sum_{n=0}^{N-1} x(nT)\,\mathrm{e}^{-\mathrm{j}k\Omega nT}\,T = F(\mathrm{j}\omega) \tag{3.21}$$

那么当$0 \leqslant t \leqslant (N-1)T$时，比较3.20式与3.21式得

$$F(\mathrm{j}\omega) = TX(k) \tag{3.22}$$

上式表明傅里叶变换分量与DFT分量通过抽样间隔建立起了联系。傅里叶变换通过对DFT乘以抽样间隔而得到。

另外要注意，在实际应用中$N \gg 1$，常常采用近似$\Omega = 2\pi/NT$，因此在本章的计算举例中做出了这一近似的假定，甚至对$N=4$也做这样的假定。

例3.3　下面通过一个简单的例子来说明3.20式的使用。计算序列{1, 0, 0, 1}的DFT，这里值得注意的是，如果实际的数据出现不连续的情况，为了计算对不连续的两端数据取平均来表示不连续点的值。将这种处理方法应用到第一个数据、最后一个数据以及其他不连续点的值。然而在求信号的频谱的时候，为了避免由于在数据的开头和结尾的数据不连续所引起的频谱失真，必须执行一个称为加窗的过程，这一主题将在第11章中讨论。在本节假定数据已经过预处理，假定这些数据表示每隔T秒记录的四个连续的电压值$x(0)=1, x(T)=0, x(2T)=0, x(3T)=1$。因此，$N=4$，对$k=0, 1, 2, 3$（由于$N-1=3$），要求计算复值$X(k)$。当$k=0$时，3.20式变成

$$\begin{aligned} X(0) &= \sum_{n=0}^{3} x(nT)\,\mathrm{e}^{-\mathrm{j}0} = \sum_{n=0}^{3} x(nT) \\ &= x(0) + x(T) + x(2T) + x(3T) \\ &= 1 + 0 + 0 + 1 = 2 \end{aligned}$$

所以$X(0)=2$是实数，幅度为2，相角$\phi(0)=0$。当$k=1$时，3.20式为

$$X(1)=\sum_{n=0}^{3}x(nT)\,e^{-j\Omega nT}$$

T还没有给定，但用$\Omega=2\pi/NT$可以消去T，于是可得

$$X(1)=\sum_{n=0}^{3}x(nT)\,e^{-j\Omega n2\pi/N\Omega}=\sum_{n=0}^{3}x(nT)\,e^{-j2\pi n/N}$$

$$=1+0+0+1e^{-j2\pi 3/4}=1+e^{-j3\pi/2}$$

$$=1+\cos\left(\frac{3\pi}{2}\right)-j\sin\left(\frac{3\pi}{2}\right)=1+j$$

因此，$X(1)=1+j$，这是一个复数，幅度为$\sqrt{2}$，相角为$\phi(\Omega)=\tan^{-1}1=45°$。当$k=2$时，3.20 式为

$$X(2)=\sum_{n=0}^{3}x(nT)\,e^{-j2\Omega nT}=\sum_{n=0}^{3}x(nT)\,e^{-j2n2\pi/N}$$

$$=\sum_{n=0}^{3}x(nT)\,e^{-j4\pi n/N}$$

$$=1+0+0+1e^{-j4\pi 3/4}=1+0+0+e^{-j3\pi}=1-1=0$$

$X(2)=0$，幅度为 0，相角$\phi(2)$不确定。最后当$k=3$时，3.20 式变成

$$X(3)=\sum_{n=0}^{3}x(nT)\,e^{-j3n2\pi/N}$$

$$=1+0+0+e^{-j9\pi/2}=1-j$$

因而$X(3)=1-j$，幅度为$\sqrt{2}$，相角为$\phi(3)=-45°$。

因此，我们证明了序列$\{1,0,0,1\}$的 DFT 是复序列$\{2,1+j,0,1-j\}$。

在实际中常常是用$|X(k)|\sim k\Omega$图以及$\phi(k)\sim k\Omega$图来表示 DFT。根据Ω的谐波或者如果Ω已知根据频率就可以实现上述操作，为了求Ω，必须知道抽样间隔T。如果假定上面的数据序列以 8 kHz 抽样，那么$T=1/(8\times10^3)=125\,\mu s$，$\Omega=2\pi/NT=2\pi/(4\times125\times10^{-6})=12.57\times10^3$弧度/秒（rad/s），因而$2\Omega=25.14\times10^3$弧度/秒，$3\Omega=37.71\times10^3$弧度/秒，图 3.3(a)画出了$x(nT)\sim t$图，图 3.3(b)画出了$|X(k)|\sim k\Omega$图，图 3.3(c)画出了$\phi(k)\sim k\Omega$图。值得注意的是，图 3.3(b)的幅度图关于二次谐波是对称的，即关于谐波数$N/2$是对称的。在图 3.3(c)中，相角是以谐波数$N/2$为中心的奇函数，这些结果具有一般性。

如果将 DFT 的第k个分量$X(k)$与第$(k+N)$个分量$X(k+N)$进行比较，我们可以推导出 DFT 的一个重要的性质，即

$$X(k)=\sum_{n=0}^{N-1}x(nT)\,e^{-jk\Omega nT}$$

$$=\sum_{n=0}^{N-1}x(nT)\,e^{-jk2\pi n/N}$$

和

$$X(k+N)=\sum_{n=0}^{N-1}x(nT)\,e^{-jk2\pi n/N}\,e^{-jN2\pi n/N}$$

$$= \sum_{n=0}^{N-1} x(nT)\,\mathrm{e}^{-\mathrm{j}k2\pi n/N}\,\mathrm{e}^{-\mathrm{j}2\pi n}$$

$$= \sum_{n=0}^{N-1} x(nT)\,\mathrm{e}^{-\mathrm{j}k2\pi n/N} = X(k)$$

由于n是整数，所以$\mathrm{e}^{-\mathrm{j}2\pi n}=1$。

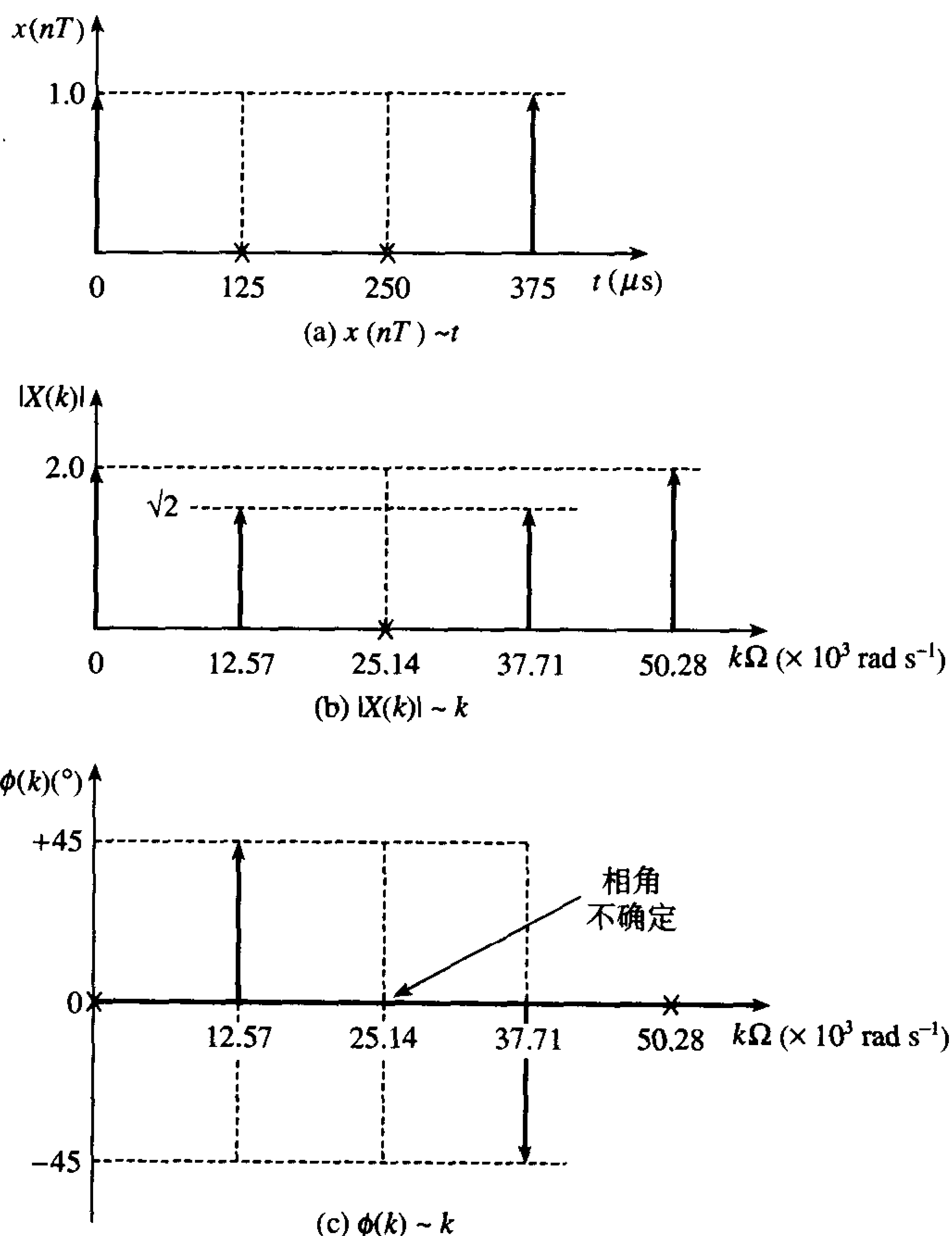


图3.3 各种示意图

$X(k+N) = X(k)$的事实表明DFT是周期的，其周期为N，这是DFT的循环特性，DFT分量的值是重复的。如果$k=0$，那么$k+N=N$，$X(0)=X(N)$。在上面的例子中，$X(0)=2$，因此$X(4)=2$。图3.3(b)说明DFT的循环特性，其中四次谐波幅度画在50.28 kHz。幅度关于二次谐波的对称性是很明显的。一般的结论是：当零次谐波和第N+1次谐波包括在图中的时候，N点DFT的幅度频谱是关于谐波$N/2$对称的。类似地，相位函数是奇函数，它展示了关于谐波$N/2$的反对称性，如果对信号每秒取$2f_{\max}$个抽样值，那么在t秒内信号被抽取的样本数为$2f_{\max}t=N$，所以，$1/t=2f_{\max}/N$是一次谐波。因此，在谐波$N/2$的对称性出现在频率$(N/2)/(2f_{\max}/N)=f_{\max}$，即信号出现的最高频率。于是，所有的信号分量都由画到$f_{\max}$即$N/2$次谐波分量的幅度谱完全表示出来，没有必要画更多的点。由于$N/2$与N之间的谐波相对于位于$f_{\max}$的对称轴可以折叠，折叠以后精确地重叠在低频那一半的频谱上，所以在本书中$f_{\max}$称为折叠频率。现在可以看出，N个实数值变换成$N/2$个复数DFT值，后者由$N/2$个实数和$N/2$个虚数给出了从原始的N个实数值推导出来的全部N个值。最后，在

例 3.3 中的数据{1, 0, 0, 1}的傅里叶变换分量 $F(\mathrm{j}\omega)$的值可由 DFT 分量乘以 $T = 125\ \mu\mathrm{s}$ 得到。因此，$F(0) = 250\ \mu\mathrm{V\ Hz^{-1}}$，$F\,(12.57\ \mathrm{kHz}) = (125+\mathrm{j}125)\ \mu\mathrm{V\ Hz^{-1}}$，$F(25.14\ \mathrm{kHz}) = 0\ \mathrm{V\ Hz^{-1}}$，$F(37.71\ \mathrm{kHz}) = (125 - \mathrm{j}125)\ \mu\mathrm{V\ Hz^{-1}}$。

正如本章引言中所解释的那样，必须也能从频域到时域进行离散变换，这可以由下面定义的离散傅里叶反变换（IDFT）来实现，

$$x(nT) = F_{\mathrm{D}}^{-1}[X(k)] = \frac{1}{N}\sum_{k=0}^{N-1} X(k)\,\mathrm{e}^{\mathrm{j}k\Omega nT},\ n = 0, 1, \ldots, N-1 \tag{3.23}$$

其中 F_{D}^{-1} 表示离散傅里叶反变换。

很明显，IDFT 与 3.16 式的傅里叶反变换类似。很容易证明，傅里叶反变换可由 IDFT 除以 T 得到。3.23 式的有效性通过将 $x(nT)$代入到 3.20 式就可以得到证明。

例3.4 从时间序列{1, 0, 0, 1}的DFT分量[2, 1 + j, 0, 1 − j]来求时间序列本身，是说明离散傅里叶反变换的很有用的一个例子。

$n = 0$ 时，

$$\begin{aligned} x(nT) = x(0) &= \frac{1}{N}\sum_{k=0}^{N-1} X(k) \\ &= \tfrac{1}{4}[X(0) + X(1) + X(2) + X(3)] \\ &= \tfrac{1}{4}[2 + (1+\mathrm{j}) + 0 + (1-\mathrm{j})] = 1 \end{aligned}$$

当 $n = 1$ 时，

$$\begin{aligned} x(nT) = x(T) &= \frac{1}{N}\sum_{k=0}^{N-1} X(k)\,\mathrm{e}^{\mathrm{j}k\Omega T} \\ &= \frac{1}{N}\sum_{k=0}^{N-1} X(k)\,\mathrm{e}^{\mathrm{j}k2\pi/N} = \frac{1}{4}\sum_{k=0}^{N-1} X(k)\,\mathrm{e}^{\mathrm{j}k\pi/2} \\ &= \tfrac{1}{4}[2 + (1+\mathrm{j})\,\mathrm{e}^{\mathrm{j}\pi/2} + 0 + (1-\mathrm{j})\,\mathrm{e}^{\mathrm{j}3\pi/2}] \\ &= \tfrac{1}{4}[2 + (1+\mathrm{j})\mathrm{j} + (1-\mathrm{j})(-\mathrm{j})] \\ &= \tfrac{1}{4}(2 + \mathrm{j} - 1 - \mathrm{j} - 1) = 0 \end{aligned}$$

当 $n = 2$ 时，

$$\begin{aligned} x(nT) = x(2T) &= \frac{1}{N}\sum_{k=0}^{N-1} x(k)\,\mathrm{e}^{\mathrm{j}k\pi} \\ &= \tfrac{1}{4}[2 + (1+\mathrm{j})\,\mathrm{e}^{\mathrm{j}\pi} + (1-\mathrm{j})\,\mathrm{e}^{\mathrm{j}3\pi}] = \tfrac{1}{4}[2 - (1+\mathrm{j}) - (1-\mathrm{j})] \\ &= 0 \end{aligned}$$

最后当 $n = 3$ 时，

$$\begin{aligned} x(nT) = x(3T) &= \frac{1}{N}\sum_{k=0}^{N-1} X(k)\,\mathrm{e}^{\mathrm{j}k3\pi/2} \\ &= \tfrac{1}{4}[2 + (1+\mathrm{j})\,\mathrm{e}^{\mathrm{j}3\pi/2} + (1-\mathrm{j})\,\mathrm{e}^{\mathrm{j}9\pi/2}] \\ &= \tfrac{1}{4}[2 + (1+\mathrm{j})(-\mathrm{j}) + (1-\mathrm{j})\mathrm{j}] = \tfrac{1}{4}(2 - \mathrm{j} + 1 + \mathrm{j} + 1) = 1 \end{aligned}$$

不出所料，得到的是序列的最后一项。

3.3　DFT 的性质

DFT 有许多性质，这些性质可以简化处理问题，或者得出有用的应用结果。下面列出这些性质，数据序列 $x(nT)$写成 $x(n)$。

(1) **对称性**

$$\mathrm{Re}\,[X(N-k)] = \mathrm{Re}\,X(k) \tag{3.24}$$

（其中 Re 表示实部）该性质描述了前面讨论过的幅度频谱的对称性。而

$$\mathrm{Im}\,[X(N-k)] = -\mathrm{Im}\,[X(k)] \tag{3.25}$$

（其中 Im 表示虚部）该性质描述了相位频谱的反对称性。在考虑频谱分量时要意识到这一性质。

(2) **偶函数**　如果 $x(n)$是偶函数 $x_e(n)$，即 $x_e(n) = x_e(-n)$，那么

$$F_D[x_e(n)] = X_e(k) = \sum_{n=0}^{N-1} x_e(n)\cos(k\Omega nT) \tag{3.26}$$

(3) **奇函数**　如果 $x(n)$是奇函数 $x_0(n)$，即 $x_0(n) = -x_0(-n)$，那么

$$F_D[x_0(n)] = X_0(k) = -\mathrm{j}\sum_{n=0}^{N-1} x_0(n)\sin(k\Omega nT) \tag{3.27}$$

(4) **Parseval 定理**　信号的归一化能量由下面的一个表达式表示，

$$\sum_{n=0}^{N-1} x^2(n) = \frac{1}{N}\sum_{k=0}^{N-1}|X(k)|^2 \tag{3.28}$$

3.28 式的右边是均方幅度谱，而左边是时间序列幅度的平方之和。

(5) **δ（delta）函数**：

$$F_D[\delta(nT)] = 1 \tag{3.29}$$

(6) **两个数据序列的互相关可以用 DFT 计算**　两个长度为 N 的有限长度序列 $x_1(n)$、$x_2(n)$的线性互相关定义为

$$r_{x_1x_2}(\mathrm{j}) = \frac{1}{N}\sum_{n=-\infty}^{\infty} x_1(n)x_2(n+\mathrm{j}),\ -\infty \leqslant \mathrm{j} \leqslant \infty \tag{3.30}$$

另外，由于循环相关（circular correlation）可以用 DFT 计算，所以定义有限长度周期序列 $x_{1p}(n)$、$x_{2p}(n)$的循环相关也是必要的。循环相关定义为

$$r_{cx_1x_2}(\mathrm{j}) = \frac{1}{N}\sum_{n=0}^{N-1} x_{1p}(n)x_{2p}(n+\mathrm{j}),\quad \mathrm{j} = 0,\ldots,N-1 \tag{3.31}$$

这样，

$$r_{cx_1x_2}(\mathrm{j}) = F_D^{-1}[X_1^*(k)X_2(k)] \tag{3.32}$$

3.32 式称为相关定理。通过加零的方法可以使由 3.32 式给出的循环相关转换成线性相关。如果 $x_1(n)$的长度为 N_1，$x_2(n)$的长度为 N_2，那么线性相关后的长度为 N_1+N_2-1。为了将循环相关转换成线性相关，给 $x_1(n)$加 N_2-1 个零后用 $x_{1a}(n)$表示，给 $x_2(n)$加 N_1-1 个零后用 $x_{2a}(n)$表示，在 3.32 式中分别用 $x_{1a}(n)$、$x_{2a}(n)$替换 $x_1(n)$、$x_2(n)$，那么 $x_1(n)$和 $x_2(n)$的线性互相关为

$$r_{x_1x_2}(j)=F_D^{-1}[X_{1a}^*(k)X_{2a}(k)] \tag{3.33}$$

其中

$$X_{1a}(k)=F_D[x_{1a}(n)] \text{ and } X_{2a}(k)=F_D[x_{2a}(n)]$$

有关相关运算将在第5章进行详细的讨论。

(7) **DFT 可以用于循环卷积的计算，补零也可以用于线性卷积的计算** 卷积可以是时域卷积，也可以是频域卷积。时域卷积定理可表述为

$$x_{3p}(n)=x_{1p}(n)\circledast x_{2p}(n)=F_D^{-1}[X_1(k)X_2(k)] \tag{3.34}$$

其中 $\circledast$ 表示循环卷积，$x_{1p}(n)$、$x_{2p}(n)$和 $x_{3p}(n)$是等长的有限周期序列。

利用类似于3.31式的方法，$x_{3p}(n)$可以写成

$$x_{3p}(n)=\sum_{m=0}^{N-1}x_{1p}(m)x_{2p}(n-m) \tag{3.35}$$

而且，

$$X_3(k)=X_1(k)X_2(k) \tag{3.36}$$

其中 $X_3(k)=F_D[x_3(n)]$。

下面的等式是频域卷积定理的表述：

$$\frac{1}{N}X_1(k)\circledast X_2(k)=F_D[x_1(n)x_2(n)] \tag{3.37}$$

其中

$$X_1(k)\circledast X_2(k)=\sum_{m=0}^{N-1}X_1(m)X_2(k-m) \tag{3.38}$$

3.34式表明时域卷积等于频域相乘，而3.37式表明频域卷积等于时域相乘。这种表述也给我们提供了记忆时域和频域关系的方法。卷积将在第5章进行详细的讨论。

3.4 DFT 计算的复杂性

计算 DFT 要求许多乘法和加法，对于8点 DFT，$X(k)$为

$$X(k)=\sum_{n=0}^{7}x(n)\,\mathrm{e}^{-\mathrm{j}k2\pi n/8},\ k=0,\ldots,7 \tag{3.39}$$

令 $k2\pi/8=K$，展开后得

$$\begin{aligned}X(k)=&x(0)\,\mathrm{e}^{-\mathrm{j}K0}+x(1)\,\mathrm{e}^{-\mathrm{j}K1}+x(2)\,\mathrm{e}^{-\mathrm{j}K2}+x(3)\,\mathrm{e}^{-\mathrm{j}K3}+x(4)\,\mathrm{e}^{-\mathrm{j}K4}\\&+x(5)\,\mathrm{e}^{-\mathrm{j}K5}+x(6)\,\mathrm{e}^{-\mathrm{j}K6}+x(7)\,\mathrm{e}^{-\mathrm{j}K7},\ k=0,\ldots,7\end{aligned} \tag{3.40}$$

3.40式的右边包含了8项，每一项由指数项（复数）与另一项（为实数或复数，例如电压时间序列为实数）相乘组成，将这些乘积项加到一起。因此，他们有8次复数乘法、7次复数加法需要计算。对于 N 点 DFT，复数乘法和复数加法的次数分别为 N 和 $N-1$。另外，有8个谐波分量需要计算（$k=0,\ldots,7$）。对于 N 点 DFT，需要计算的谐波分量有 N 个，因此，8点 DFT 的计算要求 $8^2=64$ 次复数乘法和 $8\times7=56$ 次复数加法。对于 N 点 DFT，需要计算的复数乘法和复数加法次数分别变

成了 N^2 和 $N(N-1)$。如果 $N = 1024$，那么大约需要一百万次复数乘法和一百万次复数加法。很显然，我们希望有一些方法来减少计算量。

如果我们注意到这些等式中有相当多的内在冗余运算，那么所要求的计算量是可以减少的。例如，如果 $k = 1$ 和 $n = 2$ 则 $e^{-jk2\pi n/8} = e^{-j\pi/2}$；如果 $k = 2$ 和 $n = 1$，则也有 $e^{-jk2\pi n/8} = e^{-j\pi/2}$。

3.5 时域抽取的快速傅里叶变换算法

在本节将要说明如何利用DFT中固有的计算冗余来减少计算量，从而加快计算速度。对于1024点DFT，要求的计算量可以减少204.8倍。能够实现这种快速运算的算法称为快速傅里叶变换，或简称为FFT。当算法在时域运用时称为时域抽取（DIT）FFT。第一个DIT算法是由Cooley和Tukey（1965）提出来的，所以也称为Cooley-Tukey算法。对时域数据进行抽取大大减少了运算量，值得注意的是节省的计算量随 $N^2-(N/2)\log_2 N$ 增加。

首先简化符号，并且建立一些数学关系。将3.20式重写为

$$X_1(k) = \sum_{n=0}^{N-1} x_n e^{-j2\pi nk/N},\ k = 0, \ldots, N-1 \tag{3.41}$$

另外因子 $e^{-j2\pi/N}$ 写为 W_N，即

$$W_N = e^{-j2\pi/N} \tag{3.42}$$

所以3.41式变成

$$X_1(k) = \sum_{n=0}^{N-1} x_n W_N^{kn},\ k = 0, \ldots, N-1 \tag{3.43}$$

这时我们可以注意一下包括 W_N 的一些关系式，首先

$$W_N^2 = (e^{-j2\pi/N})^2 = e^{-j2\pi 2/N} = e^{-j2\pi/(N/2)} = W_{N/2} \tag{3.44}$$

其次，

$$\begin{aligned} W_N^{(k+N/2)} &= W_N^k W_N^{N/2} = W_N^k e^{-j(2\pi/N)(N/2)} = W_N^k e^{-j\pi} \\ &= -W_N^k \end{aligned} \tag{3.45}$$

为了方便起见，总结 W_N 的一些有用的结果如下：

$$W_N = e^{-j2\pi/N} \tag{3.46a}$$

$$W_N^2 = W_{N/2} \tag{3.46b}$$

$$W_N^{(k+N/2)} = -W_N^k \tag{3.46c}$$

为了利用由3.46式表示的计算冗余，将数据序列分成两个等长的序列，一个是偶数序列，另一个为奇数序列。对于两个等长序列而言，数据个数肯定为偶数。如果原始数据序列数据个数为奇数，那么可以加一个零使数据个数变为偶数。这就允许我们将DFT $X_1(k)$写成两个DFT $X_{11}(k)$和$X_{12}(k)$，这两个DFT分别是偶数序列和奇数序列的DFT（参见表3.1）。这样，N点的DFT就转换成了两个$N/2$点的DFT，重复这一过程直到将$X_1(k)$分解成$N/2$个2点的DFT，这些点都是原始数据。因此，在实际中，原始数据被重新整序，$N/2$个2点DFT取一对数据进行计算。这些DFT的输出每4个组合在一起，提供$N/4$个4点DFT进行计算。然后再进行适当的组合计算8点的DFT，不断进行下去直到最后得到N点DFT $X_1(k)$。这个过程证明如下。

在3.43式中，下标n从0扩展到N-1，对应于数据$x_0, x_1, x_2, x_3,..., x_{N-1}$，偶数序列是$x_0, x_2, x_4,..., x_{N-2}$，奇数序列是$x_1, x_3,..., x_{N-1}$，两个序列都包含$N$/2点数据。偶数序列项可以表示为$x_{2n}$（$n = 0 \sim N/2-1$），奇数序列表示为$x_{2n+1}$，那么3.43式可重写为

$$\begin{aligned} X_1(k) &= \underbrace{\sum_{n=0}^{N/2-1} x_{2n} W_N^{2nk}}_{\text{偶数序列}} + \underbrace{\sum_{n=0}^{N/2-1} x_{2n+1} W_N^{(2n+1)k}}_{\text{奇数序列}} \\ &= \sum_{n=0}^{N/2-1} x_{2n} W_N^{2nk} + W_N^k \sum_{n=0}^{N/2-1} x_{2n+1} W_N^{2nk}, \quad k = 0, \ldots, N-1 \end{aligned} \tag{3.47}$$

利用3.46b式得$W_N^{2nk} = W_{N/2}^{nk}$，所以3.47式变成

$$X(k) = \sum_{n=0}^{N/2-1} x_{2n} W_{N/2}^{nk} + W_N^k \sum_{n=0}^{n=N/2-1} x_{2n+1} W_{N/2}^{nk}, \quad k = 0, \ldots, N-1 \tag{3.48}$$

3.48式可以写成

$$X_1(k) = X_{11}(k) + W_N^k X_{12}(k), \quad k = 0, \ldots, N-1 \tag{3.49}$$

将3.49式与3.43式进行比较，可以看出，$X_{11}(k)$确实是偶数序列的DFT，而$X_{12}(k)$是奇数序列的DFT。因此，正如前面所描述的那样，$X_1(k)$可以用两个DFT $X_{11}(k)$和$X_{12}(k)$来表示。在$X_{11}(k)$和$X_{12}(k)$中都会出现因子$W_{N/2}^k$，而它只需要计算一次。

表3.1说明了8点DFT的过程，第一行给出了数据，而第二行给出了数据序列的DFT。数据序列的DFT是根据偶数序列的DFT $X_{11}(k)$和奇数序列的DFT $X_{12}(k)$来表示的。第三行给出了重新整序后的数据，这些数据分别来自于计算$X_{11}(k)$和$X_{12}(k)$的数据（即偶数序列和奇数序列）。第四行给出了第三行数据序列的DFT，DFT是用第三行A_1（或A_2）偶数序列的DFT和A_1（或A_2）奇数序列的DFT来表示，分别记为$X_{21}(k)$（或$X_{23}(k)$）、$X_{22}(k)$（或$X_{24}(k)$）。第五行给出了这些DFT用到的数据序列，可以看出，这些序列最终是2点序列，它们的DFT是$X_{21}(k)$、$X_{22}(k)$、$X_{23}(k)$）、$X_{24}(k)$，这些DFT用第六行中的数据来表示。这样，一个8点的DFT可以分解成4个2点的DFT，每一个DFT得到2个值，例如$X_{21}(k)$有2个值$X_{21}(0)$和$X_{21}(1)$。这一过程包含了2次分解，在每一步将加权W_N^k取平方。查看一下第六行，可以看出

表3.1　8点FFT的结构

行号	行内容									k的范围	N的范围
1	数据序列A_0	A_0	$x_0\ x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7$								0,..., 7
2	A_0的8点DFT	$X_1(k) = X_{11}(k) + W_N^k X_{12}(k)$								0,..., N-1 (0,..., 7)	0,..., 7
3	将A_0重新整序为2个序列A_1和A_2	A_1	$x_0\ x_2\ x_4\ x_6$			A_2	$x_1\ x_3\ x_5\ x_7$				0,..., 3
4	A_1和A_2的4点DFT	$X_{11}(k) = X_{21}(k) + W_{N/2}^k X_{22}(k)$				$X_{12}(k) = X_{23}(k) + W_{N/2}^k X_{24}(k)$				0,..., N/2-1 (0,..., 3)	0,..., 3
5	将A_1和A_2重新整序为4个序列A_3, A_4, A_5, A_6	A_3	$x_0 x_4$	A_4	$x_2 x_6$	A_5	$x_1 x_5$	A_6	$x_3 x_7$		0, 1
6	A_3, A_4, A_5, A_6的2点DFT	$X_{21}(k) = x_0 + W_{N/4}^k x_4$		$X_{22}(k) = x_2 + W_{N/4}^k x_6$		$X_{23}(k) = x_1 + W_{N/4}^k x_5$		$X_{24}(k) = x_3 + W_{N/4}^k x_7$		0,..., N/4-1 (0, 1)	0, 1

$$X_{21}(k) = x_0 + W_{N/4}^k x_4 \quad k = 0, \ldots, N/4-1, \ k = 0, 1 \tag{3.50}$$

于是

$$X_{21}(0) = x_0 + x_4$$

而

$$\begin{aligned} X_{21}(1) &= x_0 + W_{N/4} x_4 \\ &= x_0 + W_2 x_4 = x_0 + \mathrm{e}^{-\mathrm{j}2\pi/2} x_4 = x_0 + \mathrm{e}^{-\mathrm{j}\pi} x_4 = x_0 - x_4 \end{aligned}$$

类似地，

$$X_{22}(0) = x_2 + x_6, \qquad X_{22}(1) = x_2 - x_6$$
$$X_{23}(0) = x_1 + x_5, \qquad X_{23}(1) = x_1 - x_5$$
$$X_{24}(0) = x_3 + x_7, \qquad X_{24}(1) = x_3 - x_7,$$

从上式可以看出，$k=1$ 与 $k=0$ 只是数据的符号不同。如果考虑 $X_{11}(k)$（$k=0,1,2,3$），这一点要予以重视，这时，

$$X_{11}(k) = X_{21}(k) + W_{N/2}^k X_{22}(k) \tag{3.51}$$

所以

$$X_{11}(0) = X_{21}(0) + W_{N/2}^0 X_{22}(0) = X_{21}(0) + X_{22}(0) \tag{3.52}$$

$$\begin{aligned} X_{11}(1) &= X_{21}(1) + W_{N/2}^1 X_{22}(1) = X_{21}(1) + \mathrm{e}^{-\mathrm{j}\pi/2} X_{22}(1) \\ &= X_{21}(1) - \mathrm{j} X_{22}(1) \end{aligned} \tag{3.53}$$

$$\begin{aligned} X_{11}(2) &= X_{21}(2) + W_{N/2}^2 X_{22}(2) = X_{21}(2) + \mathrm{e}^{-\mathrm{j}(2\pi/8)2\times 2} X_{22}(2) \\ &= X_{21}(2) + \mathrm{e}^{-\mathrm{j}\pi} X_{22}(2) = X_{21}(2) - X_{22}(2) \end{aligned} \tag{3.54}$$

而

$$X_{21}(2) = x_0 + W_{N/4}^2 x_4 = x_0 + W_2^2 x_4 = x_0 + x_4 = X_{21}(0)$$

和

$$X_{22}(2) = x_2 + W_{N/4}^2 x_6 = x_2 + x_6 = X_{22}(0)$$

因此，3.54 式等价于

$$X_{11}(2) = X_{21}(0) - X_{22}(0) \tag{3.55}$$

$$X_{11}(3) = X_{21}(3) + W_{N/2}^3 X_{22}(3) \tag{3.56}$$

而

$$\begin{aligned} X_{21}(3) &= x_0 + W_{N/4}^3 x_4 = x_0 + \mathrm{e}^{-\mathrm{j}(2\pi/2)3} x_4 \\ &= x_0 + \mathrm{e}^{-\mathrm{j}3\pi} x_4 = x_0 - x_4 = X_{21}(1) \end{aligned}$$

和

$$X_{22}(3) = x_2 - x_6 = X_{22}(1)$$

因此，3.56 式等价于

$$\begin{aligned} X_{11}(3) &= X_{21}(1) + \mathrm{e}^{-\mathrm{j}(2\pi/4)3} X_{22}(1) = X_{21}(1) + \mathrm{e}^{-\mathrm{j}3\pi/2} X_{22}(1) \\ &= X_{21}(1) + \mathrm{j} X_{22}(1) \end{aligned} \tag{3.57}$$

将这些结果整理在一起，得

$$X_{11}(0)=X_{21}(0)+X_{22}(0)=X_{21}(0)+W_8^0X_{22}(0) \tag{3.58a}$$

$$X_{11}(2)=X_{21}(0)-X_{22}(0)=X_{21}(0)-W_8^0X_{22}(0) \tag{3.58b}$$

$$X_{11}(1)=X_{21}(1)-\mathrm{j}X_{22}(1)=X_{21}(1)+W_8^2X_{22}(1) \tag{3.58c}$$

$$X_{11}(3)=X_{21}(1)+\mathrm{j}X_{22}(1)=X_{21}(1)-W_8^2X_{22}(1) \tag{3.58d}$$

观察一下这些方程，可以看出DFT $X_{11}(k)$是如何与偶数序列和奇数序列的DFT建立起联系的。$X_{11}(0)$和$X_{11}(2)$有公共项，只是符号不同，$X_{11}(1)$和$X_{11}(3)$也是如此。这些方程称为分解方程，因为从数据对开始，形成了$X_{21}(k)$、$X_{22}(k)$、$X_{23}(k)$、$X_{24}(k)$，允许求出$X_{11}(k)$和$X_{12}(k)$，因此也就求出了$X_1(k)$。通过这样的方法，所包含的复数加法和乘法运算的次数得以减少，因为(i)分解方程是根据循环因子W_N表示的；(ii)利用了$X_{21}(2)=X_{21}(0)$和$X_{21}(3)=X_{21}(1)$这种类型的关系；(iii)每一对表达式利用了只是符号有差别的特点。这一算法就称为Cooley-Tukey算法。

3.5.1 蝶形运算

利用符号不同为中心的对称性，取一对方程，3.58式可以用图形表示。由方程3.58a式和3.58b式，重新分解的输出是$X_{11}(0)$和$X_{11}(2)$，它们是由输入$X_{21}(0)$和$X_{22}(0)$形成的，如图3.4(a)所示。输入在交叉线的左边，输出在右边。图3.4(b)用图解的方法说明了如何得到输出$X_{11}(1)$和$X_{11}(3)$。将图3.4(a)和3.4(b)重叠在一起就得到了分解图，其中输出按k的增加的顺序排列，如图3.4(c)所示。图3.4(a)或3.4(b)的结构称为蝶形（butterfly）运算。图3.5利用这种方法描述了整个8点FFT。

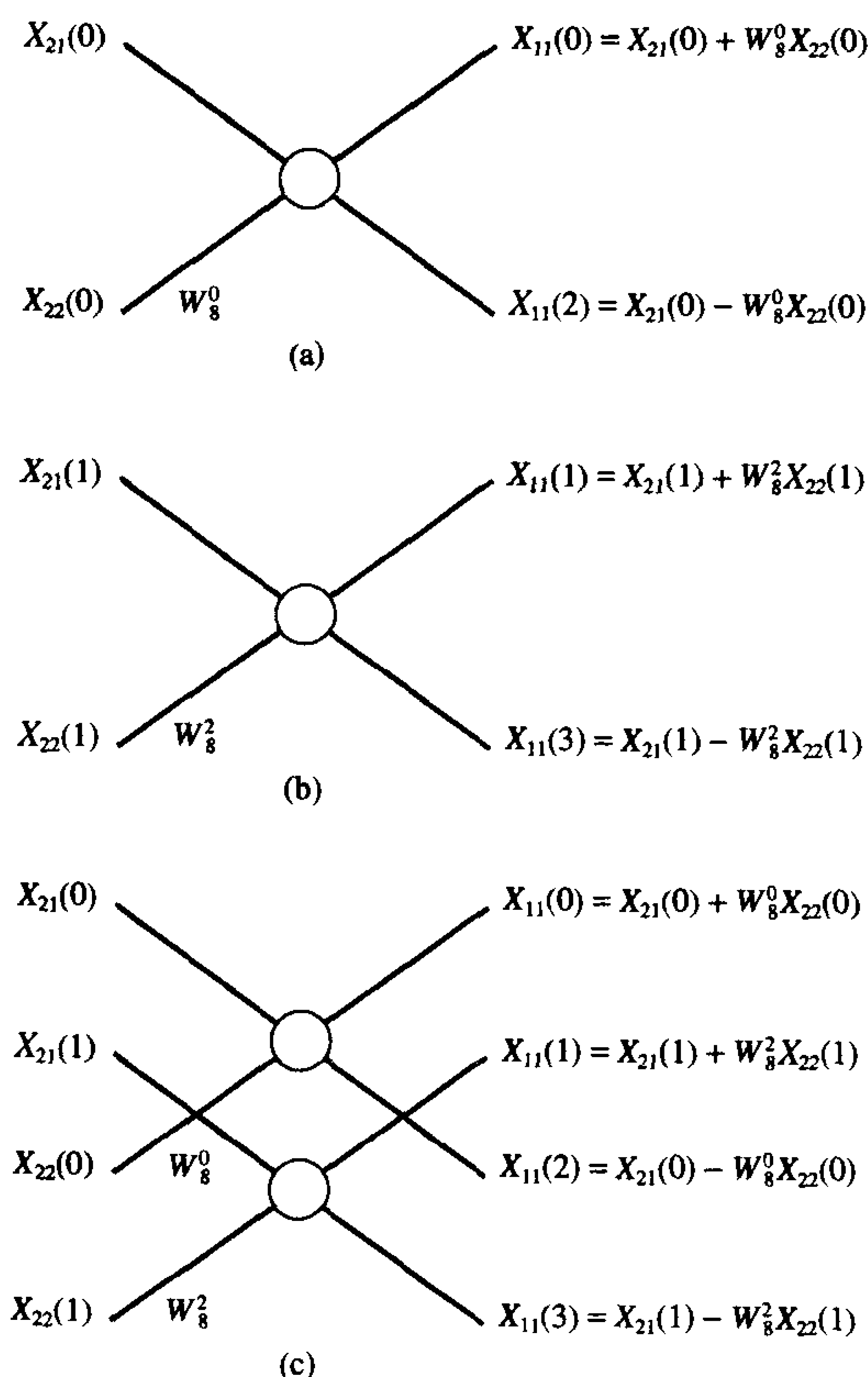


图3.4 FFT蝶形运算

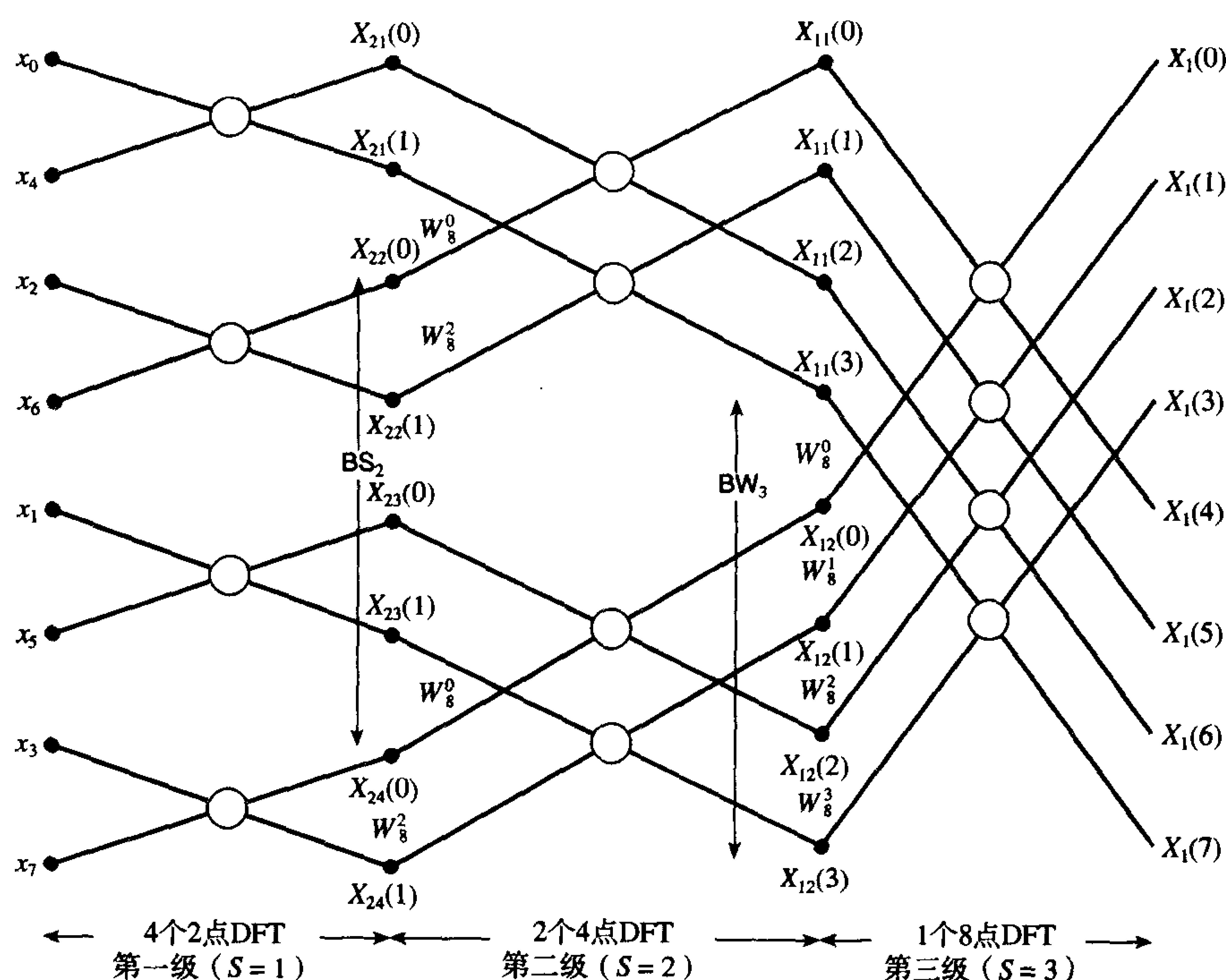


图3.5 8点DFT的FFT蝶形运算：BW_3，贡献给第三级最上端点之间的存储器间隔；BS_2，第二级中具有相同加权因子的蝶形运算底部点之间的存储器间隔

例3.5 用时域抽取FFT算法求3.2节计算过的序列{1, 0, 0, 1}的DFT，将会对我们有所启发。这是一个4点的DFT，$x_0=1, x_1=0, x_2=0, x_3=1, X_1(k)=X_{11}(k)+W_N^k X_{12}(k)$（$k=0,1,2,3$），重新整序序列为$x_0, x_2, x_1, x_3$。

我们现在用图3.5左上角的蝶形来计算DFT。点x_0, x_4, x_2, x_6用x_0, x_2, x_1, x_3取代，要求的DFT值为$X_{11}(0)$、$X_{11}(1)$、$X_{11}(2)$和$X_{11}(3)$，所以，

$$X_{21}(0)=x_0+x_2=1$$

$$X_{21}(1)=x_0-x_2=1$$

$$X_{22}(0)=x_1+x_3=1$$

$$X_{22}(1)=x_1-x_3=-1$$

$$X_{11}(0)=X_{21}(0)+W_8^0X_{22}(0)=1+1=2$$

$$X_{11}(1)=X_{21}(1)+W_8^2X_{22}(1)=1+\mathrm{e}^{-\mathrm{j}\pi/2}(-1)=1+\mathrm{j}$$

$$X_{11}(2)=X_{21}(0)-W_8^0X_{22}(0)=1-1=0$$

$$X_{11}(3)=X_{21}(1)-W_8^2X_{22}(1)=1-\mathrm{j}$$

这些值与3.2节求出的值是一样的，但用FFT更容易求出。这一结论具有一般性，随着数据量的增加，节省的计算量也相应增加。

3.5.2 算法开发

考察图3.5可以看出，为了执行FFT，程序必须对输入数据进行整序和执行蝶形运算。现在我们依次进行讨论（也可以参见Strum and Kirk, 1988）。

3.5.2.1 对输入数据的重新整序

初略地一看，似乎没有明显的方法可以对输入数据的重新整序进行编程，秘诀在于要用二进制项来考虑问题。表3.2第一列给出了图3.5中输入到蝶形网络的要求整序的数据，假定每个数据以二进制存储地址进行存储，这些地址在第二列中给出，第三列给出了这些地址的位倒序地址。如果对原始数据序列地址按位倒序，$x(0)$从000开始，那么对应的位倒序地址在第四列给出，可以看出第四列包含了原始数据序列的地址。于是，我们看到重新整序数据的地址是原始数据的地址位倒序地址。因此，要求程序将数据序号（0~N−1）转换成二进制，将这些二进制按位倒序，然后再转回成十进制数，即重新整序数据的地址。转换成二进制可以通过反复除2取余数来实现。在高级语言中，余数可以用MOD函数实现，例如MOD(K, 2)给出了十进制数K除2的余数，$K/2$的整数部分可用整数除法得到，重复这一过程可求得剩余的数，直到做了$\log_2 N$次除法。这是因为数据是由$2^m = N$个数据点组成，所以每一个地址要求m个数，要被2除m次，其中$m=\log_2 N$。新地址的第l位是2^{m-1-l}的二进制系数，所以对于某个K（K = IADDR）值，新地址（NADDR）可以用DO循环求得，其中循环变量I = 0到I = m−1，新地址通过反复将K = IADDR除2取得的余数（RMNDR）来得出。这一计算过程的伪码为

```
DO FOR I=0 TO m–1
      RMNDR:=MOD(IADDR,2)
      NADDR:=NADDR+RMNDR×2^(m-1-I)
      IADDR:=IADDR/2
END DO
```

这一DO循环必须嵌套在另一个DO循环中，其功能是从复数据组DATA(K)（K = 0... N−1）提出原始数据，K是数组中复数元素的序号，即对应于数据的原始地址。将重新整序后的数据插入到数组NEWDATA(NADDR)中，在NEWDATA中的数据就是蝶形运算中正确的数据序列，完整的伪码为

```
DO FOR K=0 TO N–1
      NADDR:=0
      IADDR:=K
      DO FOR I=0 TO m–1
            RMNDR:=MOD(IADDR,2)
            NADDR:=NADDR+RMNDR×2^(m-1-I)
            IADDR:=IADDR/2
      END DO
      NEWDATA(NADDR):=DATA(K)
END DO
```

表3.2 以位倒序对输入数据重新整序

蝶型运算要求整序的序列	数据的二进制地址	位倒序地址	对应的序列 = 原始数据序列
x_0	000	000	x_0
x_4	100	001	x_1
x_2	010	010	x_2
x_6	110	011	x_3
x_1	001	100	x_4
x_5	101	101	x_5
x_3	011	110	x_6
x_7	111	111	x_7

3.5.2.2 蝶形计算

计算包括三个步骤：

(1) 计算加权因子，$W_N^R = e^{-j(2\pi/N)R}$；

(2) 计算每一级的蝶形（每级的定义参见图3.5）；

(3) 计算所有的级。

有效的方法就是计算包含在每级的加权因子 W_N^R，在每级中具有相同因子的蝶形都采用这个因子进行蝶形运算。在该级的所有蝶形计算都计算完以后，在下一级再重复这一过程，直到所有的级都计算完为止。参见图3.5中的第二级 W_8^0，其中对含有 W_8^0 的两个蝶形进行了运算。接着进行含有 W_8^2 的2个蝶形运算。从第一级开始，每一级依次执行这一运算过程，并且对数据进行重新整序。为了用相对短的程序进行FFT计算，要求算法的许多性质。令蝶形宽度BWIDTH表示参与蝶形运算的2点之间的存储间隔。对于图3.5中第三级底部的蝶形，这个存储间隔是 BW_3。它有4个点的间隔，考察其他级的蝶形，可以得出一般的结论，

$$BWIDTH=2^{S-1} \tag{3.59}$$

其中S是级号(BWIDTH−1)是加权因子指数改变的步骤号。令蝶形间隔BSEP是级中具有相同加权因子的最近蝶形之间点的存储地址间隔。在图3.5中的第二级，BS_2 表示具有相同加权因子 W_8^0 的蝶形的BSEP。考察一下该图就会发现，对于第S级，

$$BSEP=2^S \tag{3.60}$$

最后，对于 N 点FFT，加权因子的指数随

$$P = N/2^S \tag{3.61}$$

改变，这可以从图3.5中可以看出。例如，在第二级，$S=2, P=8/2^2=2$，所以加权因子是 W_8^0 和 W_8^2。

每一个蝶形可以如下计算：

$$\begin{aligned}&XNEW(TOP)=XOLD(TOP)+W_N^R\times XOLD(BOTTOM)\\&XNEW(BOTTOM)=XOLD(TOP)-W_N^R\times XOLD(BOTTOM)\end{aligned} \tag{3.62}$$

其中左边称为蝶形的输入，右边称为蝶形的输出。如果按照下面的方式重写这些方程，那么可以节省存储器空间，

$$TEMP=W_N^R\times X(BOTTOM)$$

其中X(BOTTOM)为前一级蝶形的输出XOLD(BOTTOM)，

$$X(BOTTOM)=X(TOP)-TEMP \tag{3.63a}$$

现在的X(BOTTOM)为要求的蝶形输出，X(TOP)为前一个值，且

$$X(TOP)=X(TOP)+TEMP \tag{3.63b}$$

其中新的左边值X(TOP)是要求的蝶形输出。

利用以上这些知识，现在就可以写出FFT的伪代码，

```
10   PI:=3.141593
20   DO FOR S=1 TO m                                    计算m级
30        BSEP:=2^S                                     计算第S级
40        P:=N/BSEP                                     (P=N/2^S)，指数变化
50        BWIDTH:=BSEP/2                                (BWIDTH=2^(S-1))蝶形输入之间的间隔
60        DO FOR J=0 TO (BWIDTH-1)                      对于某一特定级计算出加权因子
70             R:=P.J                                   计算W_N的幂
80             THETA:=2×PI×R/N                          计算e^-J的指数
90             WN:=CMPLX{cos(THETA),-sin(THETA)}        计算W_N^R
100            DO FOR TOPVAL=J STEP BSEP UNTIL N/2      对级中所有的第J个蝶形
110                 BOTVAL:=TOPVAL+BWIDTH
120                 TEMP:=X(BOTVAL)×WN
130                 X(BOTVAL):=X(TOPVAL)-TEMP
```

```
140                 X(TOPVAL):=X(TOPVAL)+TEMP
150            END DO
160       END DO
170  END DO
```

100行～150行计算级中具有相同加权因子的所有的蝶形，在一级中总是存在$N/2$个蝶形。

3.5.3 FFT的计算优势

FFT的计算优势可以通过考虑图3.5的FFT算法来加以说明。这个图形表明N点的FFT在每级包含$N/2$个蝶形和$\log_2 N$级，即包含有$(N/2)\log_2 N$个蝶形。图3.4(a)表明每个蝶形包含有形式为$W_N^R X_{ij}(k)$的复数乘法。因此，FFT包含有$(N/2)\log_2 N$个复数乘法，而DFT正如3.4节所表明的那样有N^2次复数乘法。这样，复数乘法的运算量节省了$N^2-(N/2)\log_2 N$。每个蝶形包含两个复数加法，所以，FFT要求$N\log_2 N$次复数加法，而DFT要求$N(N-1)$次复数加法。因此，在复数加法中的运算量节省了$N(N-1)-N\log_2 N$。节省的运算量如表3.3所示。对于一个典型的1024点DFT运算量，如果采用FFT，运算量将减少两个数量级。

表3.3 当用FFT来代替DFT时节省的复数乘法和加法运算

N	DFT 复数乘法次数	DFT 复数加法次数	FFT 复数乘法次数	FFT 复数加法次数	DFT的乘法次数与FFT乘法次数之比	DFT的加法次数与FFT加法之比次数
2	4	2	1	2	4	1
4	16	12	4	8	4	1.5
8	64	56	12	24	5.3	2.3
16	256	240	32	64	8.0	3.75
32	1024	992	80	160	12.8	6.2
64	4096	4032	192	384	21.3	10.5
128	16 384	16 256	448	896	36.6	18.1
256	65 536	65 280	1024	2048	64.0	31.9
512	262 144	261 632	2304	4608	113.8	56.8
1024	1 048 576	1 047 552	5120	10 240	204.8	102.3
2048	4 194 304	4 192 256	11 264	22 528	372.4	186.1
4096	16 777 216	16 773 120	24 576	49 152	682.7	341.3
8192	67 108 864	67 100 672	53 248	106 496	1260.3	630.0

3.6 快速傅里叶反变换

求快速傅里叶反变换的FFT算法是很容易从FFT算法得到的，它可应用于将谱变换到它对应的波形，或者通过使用基本相同的算法得到原始数据来检查已经计算出的FFT是否正确。为了看清IFFT是如何推导的，在3.20式中将做如下的替换，对变量λ而不是变量n求和，令变量k变成μ，令$\Omega=2\pi/NT$。这样，e的指数变成了$-jk(2\pi/N)\lambda$，再利用符号表示$x(\lambda T)=x(\lambda)$等，那么3.20式变成

$$X(\mu)=\sum_{\lambda=0}^{N-1}x(\lambda)\,e^{-j\mu(2\pi/N)\lambda}\quad \mu=0,1,\ldots,N-1 \tag{3.64}$$

在3.23式中做类似的替换，即令$k=\lambda$、$n=\mu$，那么3.23式变成

$$x(\mu)=\frac{1}{N}\sum_{\lambda=0}^{N-1}X(\lambda)\,e^{j\lambda(2\pi/N)\mu}\quad \mu=0,1,\ldots,N-1 \tag{3.65}$$

在最后两个方程中，$X(\mu)$、$X(\lambda)$、$x(\lambda)$和$x(\mu)$是等维数组X和x的所有元素，所以，可以看出IFFT $x(\mu)$只有一个因子$1/N$和指数的符号不同于FFT $X(\mu)$。因此，只要进行很小的修改，FFT就可以用来计算IFFT。通过对前面的伪码做下列修改，两个变换可以包含在同一算法中，

```
line 5      K:=1 FOR FFT, K:=-1 FOR IFFT
line 80     THETA:=K×2×PI×R/N
line 145    IF K=-1 DO
line 146    X(BOTVAL):=X(BOTVAL)/N
line 147    X(TOPVAL):=X(TOPVAL)/N
line 148    END DO
```

3.7 FFT的实现

现在，在原理上应该是清楚的，FFT或者IFFT通过提供一个数组，并且利用FFT或IFFT算法（包括位倒序算法）对数据进行操作来计算的。然而，其中还存在一些其他考虑。到目前为止，数据两端出现的不连续所带来的影响，以及称为混淆（aliasing）和栅栏（picket fencing）的效应都被忽略。为了对真实数据谱求得一个好的近似，必须使用第11章描述的技术来考虑这些影响。另一方面在于这样一个事实，到目前为止，本章关注的是基-2时域抽取算法，但是还存在一些其他算法，包括频域抽取（DIF）。对于其中的一些问题将在下面进行讨论。

3.7.1 频域抽取FFT

3.5节描述了时域抽取快速傅里叶变换算法，这种算法是将原始的3.43式的离散傅里叶变换反复地划分成两个变换而得到的。一个变换由偶数项组成，另一个变换由奇数项组成，一直到原始的变换被化成原始数据的2点变换为止。另一种方法是将原始的变换分成两个变换，一个包含有前一半数据，另一个包含有后一半数据，由此得到频域抽取算法。这一算法最初是由Gentleman and Sande(1966)推导的，通常称为Sande-Tukey算法。总地来说，两种算法之间几乎不存在选择的问题。

3.7.2 DIT和DIF算法的比较

对于DIF算法而言，输入数据的顺序是不变的，而输出FFT序列是位倒序的。DIT和DIF两种算法都是原位算法（in-place algorithm）。重画算法图、维持输入输出数据的顺序是可能的，但得到的算法就不是原位算法，并且要求额外的存储空间（参见第12章）。两种算法要求的复数乘法次数是相同的。总地来说，两种算法之间几乎不存在选择的问题。

3.7.3 增加运算速度的修改

对于DIT算法而言，进一步提高计算速度是可能的。例如，用基-4 FFT可以进一步减少复数乘法次数接近2倍，加法运算次数也将减少。另一种提高速度的方法是：当加权因子$W_N=\pm1$或者$\pm j$时，通过加权因子W_N（常常也称为旋转因子）可以消去不必要的乘法，这同样也可以减少要求的加法次数。对于$W_N=\pm1$或者$\pm j$的情况，是通过包含一个分离的蝶形来实现的。例如，我们给出一个基-2的双蝶形原位DIT算法。首先计算W_N的正弦和余弦部分，然后将它们存在一个查找表中，要用到它们的值时就从表中查出，这样也可以进一步节省运算时间。还有一些其他的改进措施，因此，显然不光是存在一个FFT，而是存在许多FFT。这些问题的理解要花费相当多的时间，并且还要有相当好的数学功底，熟悉先进的矩阵理论、多维指数映射和数论是有优势的。这些综合处理方法超出了本章的目标，我们是以数字信号处理领域的大多数人能够理解的方法来解释离散变换的

概念。有许多其他的书籍，例如Burrus and Parks(1985)、Beauchamp(1987)介绍了许多特殊的方法，推荐读者去阅读这些书籍。正如已经指出的那样，有许多算法可供选择，专家可能写出了他自己的适合于应用的算法。然而，减少乘法和加法运算并不总是能提高运算速度。乘法的减少可能增加更多的程序代码和更多的加法。如果采用硬件处理芯片，那么可能会加入一些限制，这些限制对算法的改善是起反作用的。由于速度的提高不可能超过2倍，因此对于一般的应用，速度不是特别关键的因素，作者倾向于推荐前面描述的基本的基-2 DIT FFT。在Burrus and Parks(1985)中给出了许多FORTRAN FFT程序，并且讨论了这些程序的优点和缺点，书中还给出了有关硬件实现的程序。本章附录3A 给出了基-2 DIT FFT 算法的C语言程序。

3.8 其他离散变换

还有许多其他有效的离散变换，Winograd 傅里叶变换（Winograd, 1978；同时参见Burrus and Parks, 1985; Beauchamp, 1987; Rader, 1968; Signal Processing Committee, 1979; McClellan and Rader, 1979）和素因子算法（Beauchamp, 1987，同时参见McClellan and Rader, 1979）提供了提高计算FFT运算速度的富有创造性的复杂算法。离散余弦变换在数据压缩应用中特别有用（参见3.8.1节）。沃尔什变换（参见3.8.2节）将信号分解成矩形波而不是正弦波，计算起来比FFT要快。由对沃尔什整序的序列重新整序构造的哈达玛变换的计算速度甚至更快。沃尔什变换和哈达玛变换在某些应用中体现其优势的同时也具有一些缺陷，这限制了它们的应用，请参见3.8.2节和3.8.3节。最后，哈尔（Haar）变换对于图像处理的边缘检测及其类似的应用特别有用（Rosenfield and Thurston, 1971）。对于那些希望知道更多内容的读者，Beauchamp(1987)提供了许多入门资料的来源。在20世纪90年代，人们对小波变换的兴趣在日益增加（Chan, 1995; Daubechies, 1990, 1992; Burrus, Gopinath and Guo, 1998），我们将在3.8.4节进行介绍。

3.8.1 离散余弦变换

变换方法除了应用于加快相关、卷积计算外，也用于数据压缩，例如语音和视频的传输以及EEG和ECG生物医学信号的记录，变换还可以用于模式识别。在这些应用中，只利用最有效的分量。因此，要求的代码位数减少，这样使传输更快，传输线的带宽更窄，也更容易进行模式识别（由于数据减少）。合适变换的三个重要特征是它的压缩效率、易于计算和最小均方误差，压缩效率与在低频段集中能量有关。满足这些特征的理想的变换是Karhunen-Loève变换，但是这种变换不能用算法表示。然而，离散余弦变换（DCT）实际上具有相同的性质，并且可以形成算法，它本质上是由DFT的实部组成的。由于实偶函数的傅里叶级数只含有余弦项，因此这一定义是合理的。例如，抽样后的电压值情况，所用的数据是实的，通过乘2再加上它的镜像（mirror image）项就可以变成对称的。这样，DFT为（3.41式）

$$X(k)=\sum_{n=0}^{N-1}x_n\mathrm{e}^{-\mathrm{j}2\pi nk/N},\quad k=0,1,\ldots,N-1$$

定义上式的实部为DCT $X_c(k)$，即

$$X_c(k)=\mathrm{Re}\,[X(k)]=\sum_{n=0}^{N-1}x_n\cos\left(\frac{k2\pi n}{N}\right),\quad k=0,1,\ldots,N-1 \tag{3.66}$$

这是几种DCT形式中的一个，更为常见的形式是（Beauchamp, 1987; Yip and Ramamohan,1987; Ahmed and Rao, 1975）

$$X_c(k)=\frac{1}{N}\sum_{n=0}^{N-1}x_n\cos\left(\frac{k2\pi n+k\pi}{2N}\right)=\frac{1}{N}\sum_{n=0}^{N-1}x_n\cos\left[\frac{k\pi(2n+1)}{2N}\right] \quad k=0,1,\ldots,N-1 \tag{3.67}$$

还存在一些其他的形式（Yip and Ramamohan, 1987; Pennebaker and Mitchell, 1993; Pitas,1993; Bailey and Birch, 1989）。

正如我们所期待的那样，DCT 是根据 FFT 实现的（Narasinka and Petersen, 1978），快速 DCT 是已经建立的 DCT 算法（Chen et al., 1977）运算速度的 6 倍。另一个版本是 C 矩阵变换，它可以通过硬件而更简单地构造（Srinivassan and Rao, 1983）。

3.8.2　沃尔什变换

目前为止所讨论的变换都是基于余弦和正弦函数，根据只取±1那样的脉冲波形的变换将更简单、计算更快，也更适合于不连续波形（比如图像）的表示。反过来，它们也不适合描述连续波形，它们也可能不是相位不变的，因此推导出来的谱可能存在失真。然而，这样的波形可以在图像处理（天文学和光谱学）、信号编码和滤波中采用。

正如 DFT 是依据一组与余弦和正弦有关的谐波一样，离散沃尔什变换（DWT）是依据一组与矩形波有关的谐波，这组矩形波称为沃尔什函数。然而，对矩形波没有定义频率，所以采用类似于序率（sequency）这样的术语。序率是每单位时间内过零的平均数的一半。图 3.6 给出了一组沃尔什函数，这组函数按序率增加的顺序一直画到 $N=8$。在时间 t 和序率 n 的沃尔什函数表示为 WAL(n, t)。考察一下图 3.6 我们可以看出，存在相等的偶数个沃尔什函数和奇数个沃尔什函数，这刚好对应于傅里叶级数的余弦分量和正弦分量。偶沃尔什函数 WAL$(2k, t)$写为 CAL(k, t)，奇沃尔什函数 WAL$(2k+1, t)$写成 SAL(k, t)，其中 $k=1, 2,\ldots, N/2-1$。

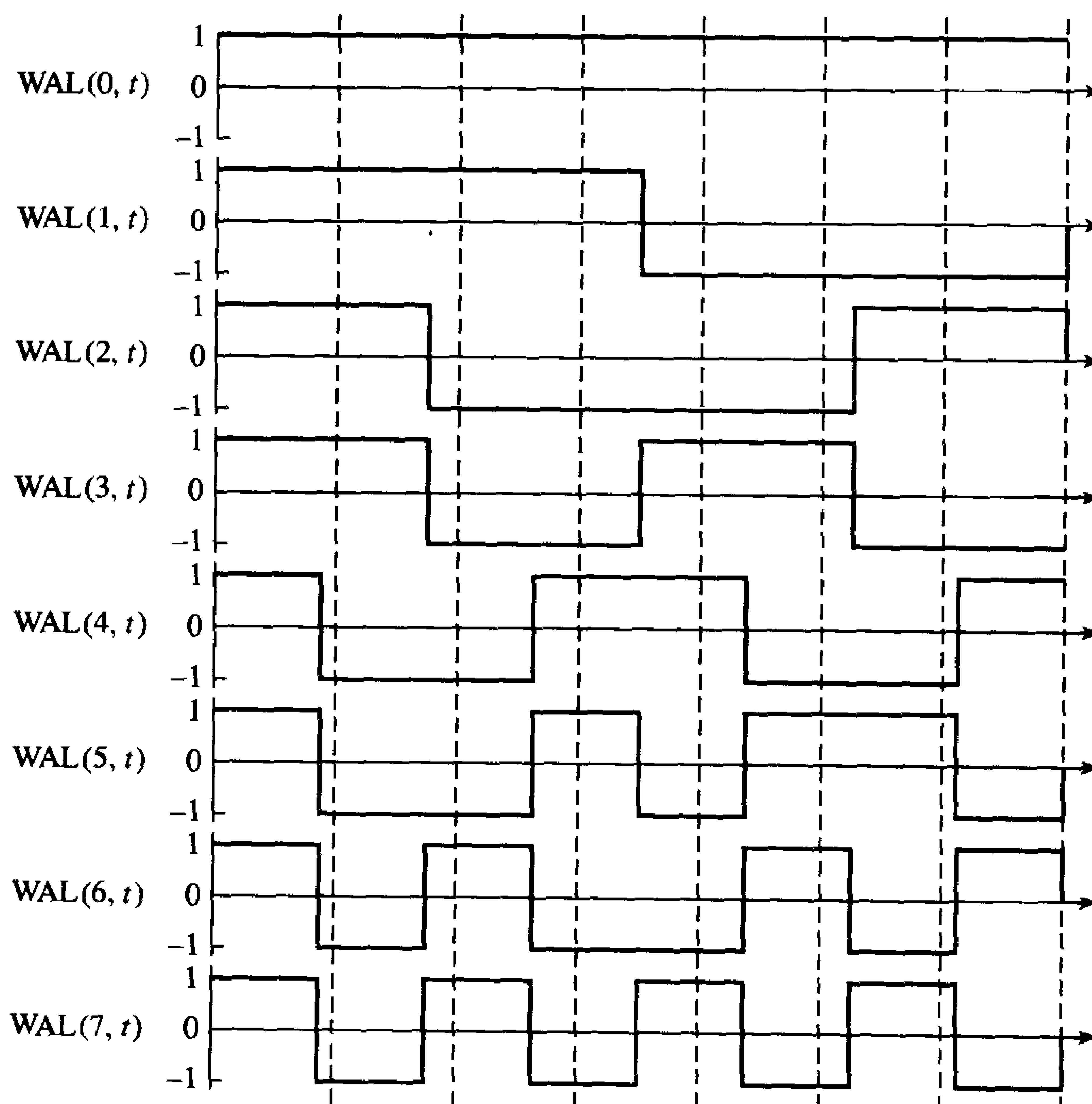


图 3.6　按序率整序一直到 $n=7$ 的沃尔什函数表明的 8 × 8 的沃尔什变换矩阵的抽样时间

类似于傅里叶级数，任何波形$f(t)$可以写成沃尔什级数，

$$f(t) = a_0 \mathrm{WAL}(0, t) + \sum_{i=1}^{N/2-1} \sum_{j=1}^{N/2-1} [a_i \mathrm{SAL}(i, t) + b_i \mathrm{CAL}(j, t)] \tag{3.68}$$

其中a_i和b_i是级数的系数。

任何两个沃尔什函数，

$$\sum_{t=0}^{N-1} \mathrm{WAL}(m, t)\, \mathrm{WAL}(n, t) = \begin{cases} N, & n = m \\ 0, & n \neq m \end{cases}$$

即沃尔什函数是正交的。

离散沃尔什变换对为

$$X_k = \frac{1}{N} \sum_{i=0}^{N-1} x_i \mathrm{WAL}(k, i) \quad k = 0, 1, \ldots, N-1 \tag{3.69}$$

和

$$x_i = \sum_{i=0}^{N-1} X_k \mathrm{WAL}(k, i) \quad i = 0, 1, \ldots, N-1 \tag{3.70}$$

其中，我们注意到除了因子$1/N$之外，逆变换与正变换是相同的，$\mathrm{WAL}(k, i) = \pm 1$。因此，变换对可以用前面提到的数字方法采用矩阵乘法进行计算。然而，缺少相位不变性意味着DWT不适合于快速相关或卷积。

3.69式说明，第k个DWT分量是由每个波形抽样值x_i乘以序率为k的沃尔什函数，然后对$k = 0$到$N-1$求和来得到的。对于所有的k个DWT分量可以用矩阵表示为

$$\mathbf{X}_K = \mathbf{x}_i \mathbf{W}_{ki} \tag{3.71}$$

其中$\mathbf{x}_i = [x_0\ x_1\ x_2 \ldots x_{N-1}]$为数据序列，

$$\mathbf{W}_{ki} = \begin{bmatrix} W_{01} & W_{02} & \cdots & W_{0,N-1} \\ W_{11} & & & \\ \vdots & & & \vdots \\ W_{N-1,1} & W_{N-1,2} & \cdots & W_{N-1,N-1} \end{bmatrix}$$

是沃尔什矩阵，$\mathbf{X}_k = [X_0\ X_1 \ldots X_{N-1}]$是DWT的$N-1$个分量。注意，$\mathbf{W}_{ki}$是$N \times N$矩阵，其中$N$是被抽样的波形点数。因此，如果有$N$个点，那么有必要考虑前$N$个序率整序的沃尔什函数，每一个函数都抽样$N$次，$\mathbf{W}_{ki}$的第$k$行对应于第$k$个序率分量的$N$个抽样值。

例3.6 举例来说，让我们计算数据序列(1, 2, 0, 3)的DWT，数据序列由四个数据点组成，所以，$\mathbf{W}_{ki}$是4×4矩阵，它可以从图3.6的前四行得出：

$$\mathbf{W}_{ki} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \tag{3.72}$$

所以，由3.71式可得

$$\mathbf{X}_k = \tfrac{1}{4}[1 \quad 2 \quad 0 \quad 3] \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} = \tfrac{1}{4}[6 \quad 0 \quad 2 \quad -4]$$

可见$X_0 = 1.5$，$X_1 = 0$，$X_2 = 0.5$，$X_3 = -1$。这比计算相应的DFT要容易得多。毫无疑问，快速DWT（FDWT）是存在的。

用功率分量可以计算出对应的谱，即

$$P(k) = [|\mathrm{CAL}(k, t)|^2 + |\mathrm{SAL}(k, t)|^2]^{1/2}$$

其中

$$\begin{aligned} P(0) &= X_c^2(0) \\ P(k) &= X_c^2(k, t) + X_s^2(k, t) \\ P\left(\frac{N}{2}\right) &= X_s^2\left(\frac{N}{2}, t\right) \end{aligned} \tag{3.73}$$

其中$k = 1, 2, ..., N/2-1$，相位分量为

$$\begin{aligned} \phi(0) &= 0, \pi \\ \phi(k) &= \tan^{-1}\left[\frac{X_s(k)}{X_c(k)}\right], \quad k = 1, 2, \ldots, N/2-1 \end{aligned} \tag{3.74}$$

和

$$\phi\left(\frac{N}{2}\right) = 2k\pi \pm \pi/2, \quad k = 0, 1, 2, \ldots$$

因此，对于上面的DWT，我们有

$$P(0) = 1.5^2 = 2.25; \phi(0) = 0, \pi$$

$$P(1) = 0^2 + 0.5^2 = 0.25; \phi(1) = \tan^{-1}\left(\frac{0}{0.5}\right) = 0$$

$$P(2) = (-1)^2 = 1; \phi(2) = \frac{\pi}{2} + 2k\pi, \quad k = 0, 1, 2$$

3.8.3 哈达玛变换

哈达玛变换或者哈达玛-沃尔什变换基本上与沃尔什变换相同,但是沃尔什函数即变换矩阵的行要重新整序。合成矩阵是由二阶的子矩阵组成的，在图3.7中，8 × 8的哈达玛矩阵表示为${}^8\mathbf{H}$，可以看出它是由矩阵

$${}^2\mathbf{H} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad 和 \quad -{}^2\mathbf{H} = \begin{bmatrix} -1 & -1 \\ -1 & 1 \end{bmatrix}$$

组成的。任何2N阶哈达玛矩阵可以从${}^2\mathbf{H}$递推得到，即

$${}^{2N}\mathbf{H} = \begin{bmatrix} {}^N\mathbf{H} & {}^N\mathbf{H} \\ {}^N\mathbf{H} & -{}^N\mathbf{H} \end{bmatrix} \tag{3.75}$$

这一递推性质是对沃尔什函数进行哈达玛整序得出的,由此得出的快速沃尔什-哈达玛变换计算起来比DWT更快。按序哈达玛（或称为自然整序）沃尔什函数如图3.8所示。哈达玛整序是由下面的沃尔什整序得到的，

(1) 用二进制表示沃尔什整序函数的序号；

(2) 对二进制数按位倒序；

(3) 将二进制数转换成格雷码（Gray code）；

(4) 将这个值转换成十进制。

$$
\begin{array}{c c}
 & i \longrightarrow \\
 & \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\
\begin{array}{c} k\ 0 \\ \downarrow\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} &
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\
1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\
1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\
1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\
1 & -1 & -1 & 1 & -1 & 1 & 1 & -1
\end{bmatrix}
\end{array}
$$

图 3.7 8 × 8 哈达玛变换矩阵

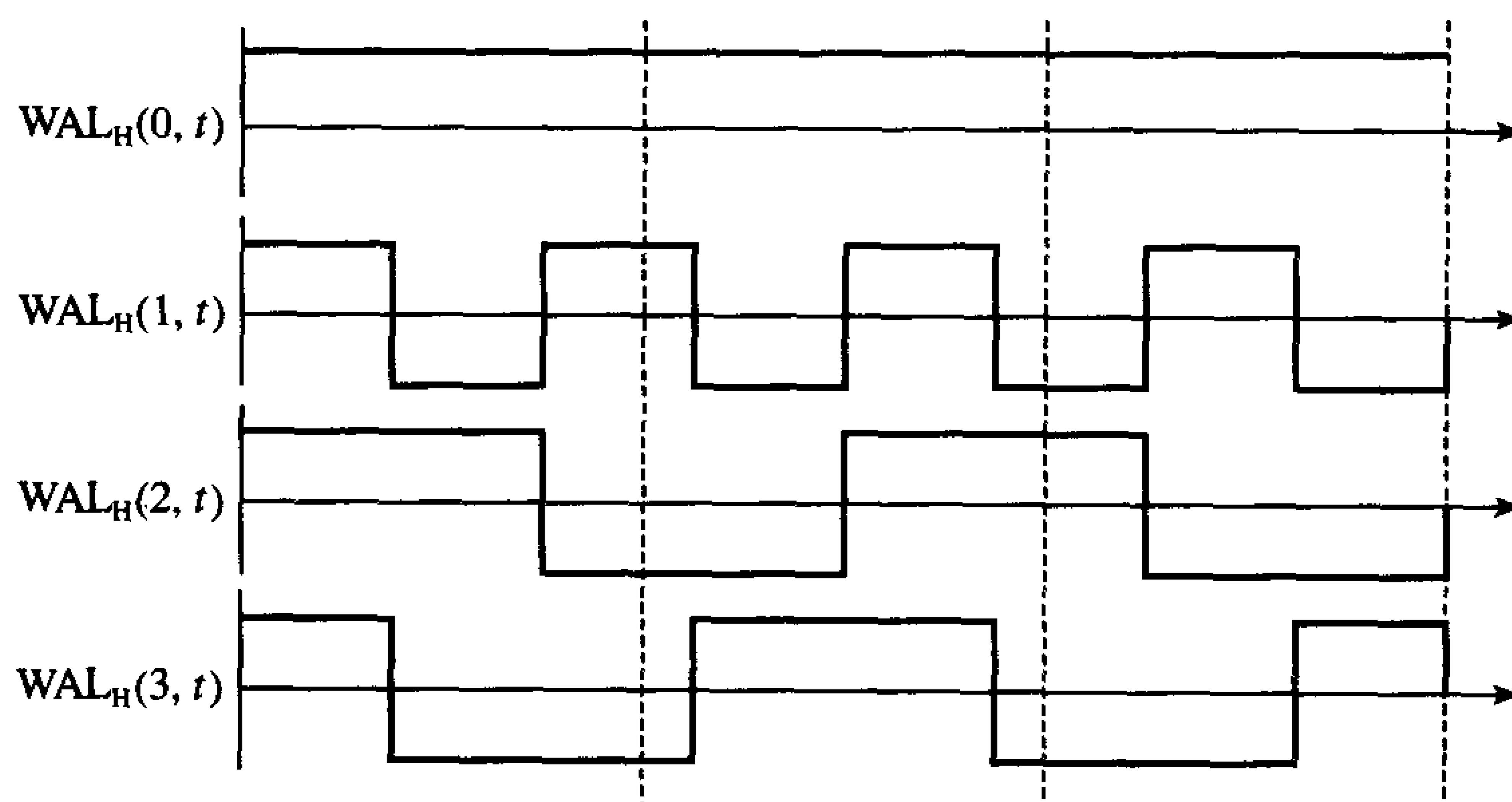


图 3.8 按序哈达玛沃尔什函数

例 3.7 举例说明序列(1, 2, 0, 3)的离散沃尔什－哈达玛变换的计算。由哈达玛矩阵的性质，4 × 4 的哈达玛矩阵 $\mathbf{H}_{ki}$ 为

$$
\mathbf{H}_{ki} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \tag{3.76}
$$

因此，(1, 2, 0, 3)的 DWHT 为

$$
\mathbf{X}_k^{\mathrm{WH}} = \tfrac{1}{4}[1 \quad 2 \quad 0 \quad 3]\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} = \tfrac{1}{4}[6 \quad -4 \quad 0 \quad 2]
$$

所以，$X_0^{\mathrm{WH}} = 1.5$，$X_1^{\mathrm{WH}} = -1$，$X_2^{\mathrm{WH}} = 0$，$X_3^{\mathrm{WH}} = 0.5$。这些沃尔什－哈达玛分量与前面计算沃尔什分量是相同的，但是被重新排序了。

3.8.4 小波变换

在物理学中的海森堡（Heisenberg）不确定性原理认为不可能同时精确地知道粒子的位置 x 和动量 p，事实上，

$$
xp \geqslant h = 6.626 \times 10^{-34}\ \mathrm{J\ s} \tag{3.77}
$$

其中 h 是普朗克（Planck）常数。利用爱因斯坦方程 $E = mc^2$，这一原理可以转化到信号处理领域，从而证明时间和频率不可能以任意精度能够同时分辨，即

$$\Delta f.T \geqslant 1 \tag{3.78}$$

其中Δf和T表示频率和时间的分辨率，如果T是高度可分辨的，那么频率就是不精确，反过来也是如此。因此，要同时以要求的精度来测量信号分量的频率和信号出现的时间，或者不同的频率分量在时间上进行分辨是不可能的。短持续时间包含高频分量的信号就是这样一种情况，短的持续时间在时间上靠得很近，长的持续时间在频率上靠得很近。这样的信号不是周期的，小波变换处理这种一般性的问题，并产生时频分析的效果，从而提供了一种分析非平稳信号的手段。小波变换也可应用于信号滤波、信号去噪以及求奇异值的位置和分布。

在傅里叶变换中，信号值是用复指数信号进行加权，复指数信号在本质上是一个正弦项。在小波变换中，信号值是用小波函数进行加权。

所有小波是从基本小波推导出来的。有许多可能的母小波，选择的母小波应该具有下列性质：它们应该是振荡的，不应该是DC分量，应该是带通的，应该随时间迅速衰减到零，应该是可逆的。最后一个性质确保了信号的小波变换是惟一的。我们可以把基本小波写成$\Psi(t)$。例如，玛里(Morlet)或修正的高斯母小波为

$$\Psi(t) = \mathrm{e}^{\mathrm{j}\omega_0 t}\,\mathrm{e}^{-t^2/2} \tag{3.79}$$

它的傅里叶变换为

$$H(\omega) = \sqrt{2\pi}\,\mathrm{e}^{-(\omega-\omega_0)^2/2} \tag{3.80}$$

这两个波形画在图 3.9 中，可以看出$\Psi(t)$是满足上面振荡和衰减的性质的。

剩余的（子）小波可由对母小波进行尺度变换形成一族小波而得到，每一个子小波可以写成

$$\frac{1}{\sqrt{a}}\Psi\{(t-\tau)/a\}$$

其中a是一个可变的尺度因子，τ是平移常数。如果尺度因子a增加，那么小波幅度和相角将减小，给定幅度，相角的减少代表频率的减少。因此，增加比例因子a对应于频率的减少，所以小波在时间水平方向展宽。正的平移因子引起小波沿正的时间轴平移。因此，通过调整尺度因子a和平移常数τ，可以产生幅度大的或小的小波、频率高的或低的小波内容，并且定位在时间上的不同位置。利用这样的方法，在不同的时间间隔扩展的具有不同的频率分量的非平稳信号可以用不同的小波和表示，这可以用小波变换实现。

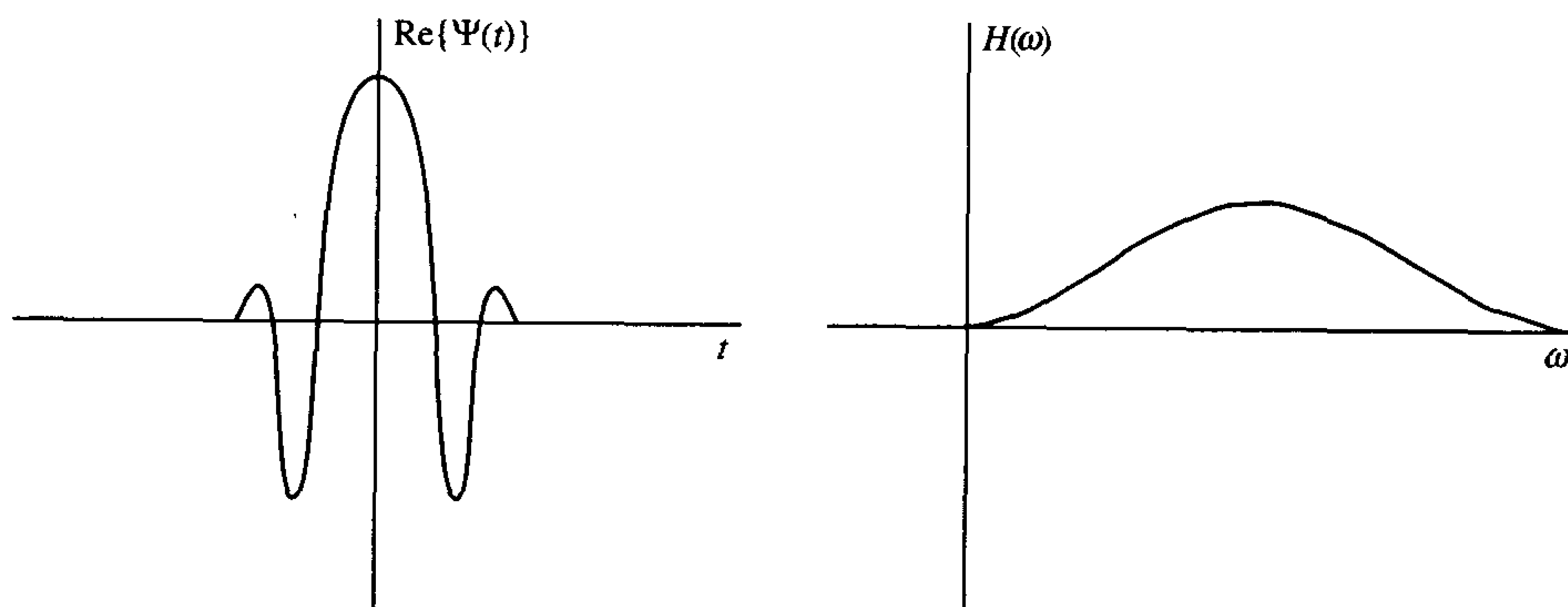


图 3.9　修正的高斯或玛里母小波 $\Psi(t)$及其傅里叶变换 $H(\omega)$

下面给出几种小波变换的定义，这些定义随着离散化程度的增加自然地发展起来，在这些定义中，假定信号$s(t)$是平方可积的，即

$$\int s^2(t)\mathrm{d}t < \infty \tag{3.81}$$

对于所有幅度有限、短持续时间的信号，这一假定是真实的。正弦和DC信号并不满足平方可积的条件。因此，在下面的讨论中将这些信号排除在外。

连续小波变换 $CWT(a, \tau)$可以定义为

$$CWT(a, \tau) = \left(1/\sqrt{a}\right)\int s(t)\Psi\{(t-\tau)/a\}\mathrm{d}t \tag{3.82}$$

式中的参数可以离散化得到离散参数的小波变换 $DPWT(m, n)$，定义为

$$DPWT(m, n) = a_0^{-m/2}\int s(t)\Psi\{(t - n\tau_0 a_0^m)/a_0^m\}\mathrm{d}t \tag{3.83}$$

其中做了下列替换：$a = a_0^m$，$\tau = n\tau_0 a_0^m$。在这些替换中，a_0 和 τ_0 是 a 和 τ 的抽样间隔，m 和 n 是整数。通常选 $a_0 = 2$，$\tau_0 = 1$。那么，

$$DPWT(m, n) = 2^{-m/2}\int s(t)\Psi\{(t - n2^m)/2^m\}\mathrm{d}t = 2^{-m/2}\int s(t)\Psi\{2^{-m}t - n\}\mathrm{d}t \tag{3.84}$$

这将时间轴展宽了 2^{-m} 倍，小波在时间上平移了 $2^m n$。

对时间离散化产生离散时间小波变换，$DTWT(m, n)$定义为

$$DTWT(m, n) = a_0^{-m/2}\sum_k s(k)\Psi(a_0^{-m}k - n\tau_0) \tag{3.85}$$

另外，如果 $a_0 = 2$，$\tau_0 = 1$，$DTWT(m, n)$变成

$$DTWT(m, n) = 2^{-m/2}\sum_k s(k)\Psi(2^{-m}k - n) \tag{3.86}$$

上式称为离散小波变换。因此，离散小波变换是从连续小波变换通过离散化尺度参数 a、平移参数 τ，并且令 $a_0 = 2$、$\tau_0 = 1$ 得到的。

除了用小波变换研究信号的时频内容之外，小波变换也可以用于信号滤波，例如消除任何出现的噪声。首先信号被变换成他们的分量；然后，辨识出噪声分量并将其消除；最后，去除了噪声的信号再从它的分量小波重构（反变换）。重构的公式为

$$s(t) = \frac{1}{C_\Psi}\int_{-\infty}^{\infty}\int_{a>0}^{\infty} CWT(a, \tau)\left\{\frac{1}{\sqrt{a}}\right\}\Psi\{(t-\tau)/a\}\left\{\frac{1}{a^2}\right\}\mathrm{d}a\mathrm{d}t \tag{3.87}$$

其中

$$C_\Psi = \int_0^{\infty}\{|H(\omega)|^2/\omega\}\mathrm{d}\omega < \infty$$

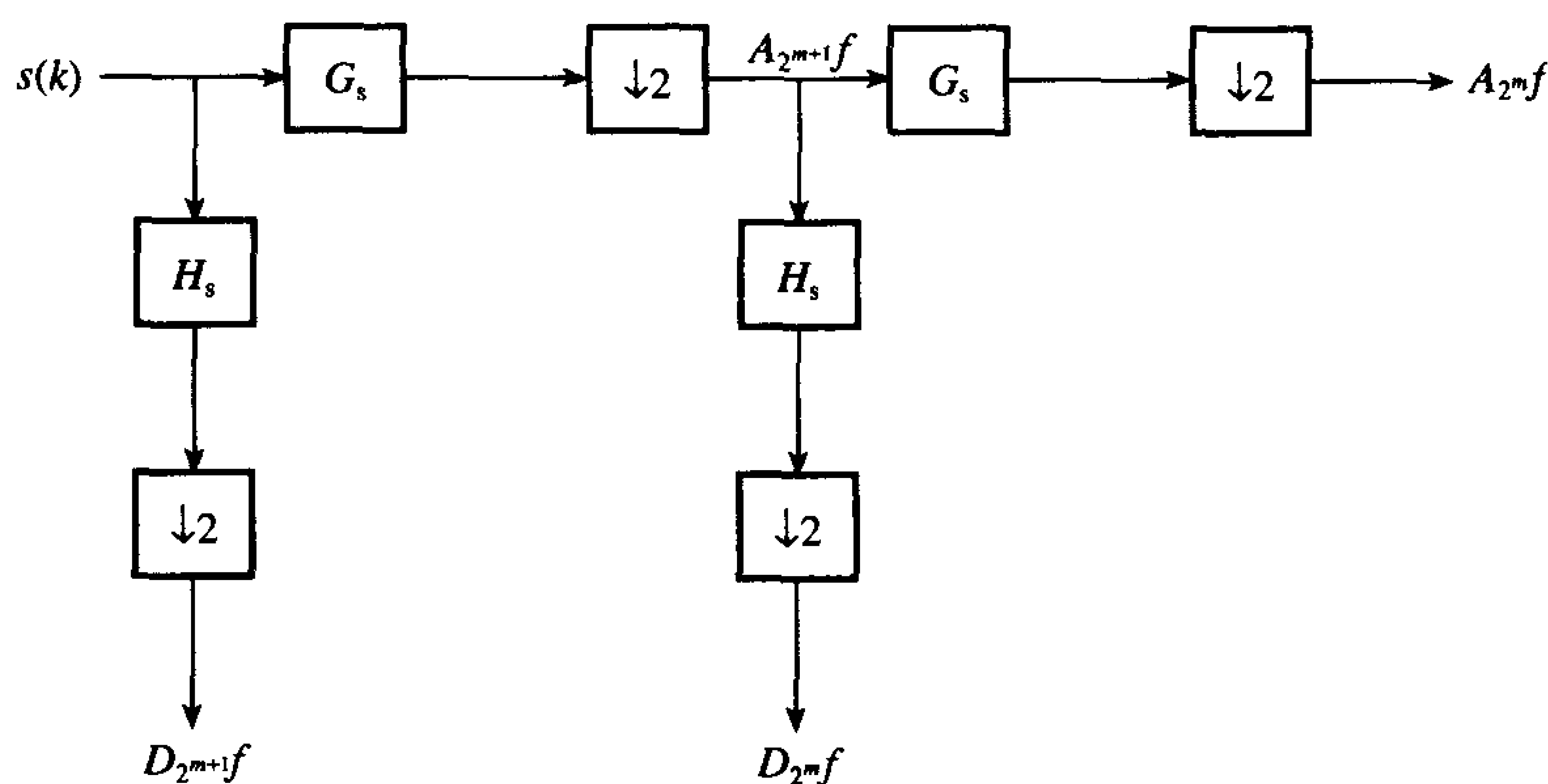
$H(\omega)$是基本小波 $\Psi(t)$的傅里叶变换。

想要了解更多的小波重构公式的读者可参考有关著作，如Chan(1995)、Daubechies(1990, 1992)、Chui(1992)、Mallat and Hwang(1992)和 Burrus(1998)。

通过考虑3.82式的连续小波变换的可能解释来理解小波变换。可以看出，一种解释是 $CWT(a,\tau)$ 表示 $s(t)$与 $\Psi(t/a)/\sqrt{a}$ 的延迟为 $-\tau/a$ 的互相关。类似地，它可能是尺度信号 $s(at)$与$\sqrt{a}\Psi(t)$在延迟为 $-\tau/a$ 的互相关。$CWT(a, \tau)$也可能表示为冲激响应为 $CWT(a, \tau)$的带通滤波器对输入信号 $s(t)$在时间 τ/a 时的输出。此外，它也可以表示冲激响应为$\sqrt{a}\Psi(-t)$的带通滤波器对输入信号 $s(at)$在时间 τ/a 时的输出。因此，小波变换可以看作为起着带通滤波器或互相关的作用。通过改变尺度因子 a，不同的频率分量被滤波。

3.8.5 利用小波变换方法的多分辨分析

多分辨分析（MRA）涉及到将信号分量划分为许多频带的问题，它可以使用低通和高通滤波器和子抽样（sub-sampling）来实现。逆过程也是可能的，即允许信号重构。MRA可以利用抽样后的信号借助离散小波变换来实现，MRA也已经应用于图像信息内容的分析（Mallat, 1989）和脑电图诱发电压的分析（Thakor et al., 1993）。Mallat(1989)发表了一种MRA，并且解释了如何应用它。倘若 $a = 2^m$，$\tau = 2^m n$，那么设计低通滤波器 $G_s(\omega)$和高通滤波器 $H_s(\omega)$是可能的。反复地应用金字塔结构将信号分解成不断增加的高分辨率频带。这一金字塔算法画在图3.10中，$G_s(\omega)$和$H_s(\omega)$分别对称于低通滤波器 $G(\omega)$和高通滤波器 $H(\omega)$。这些滤波器是正交镜像滤波器，与小波函数有关，用图3.11所示的金字塔算法可实现信号的重构。



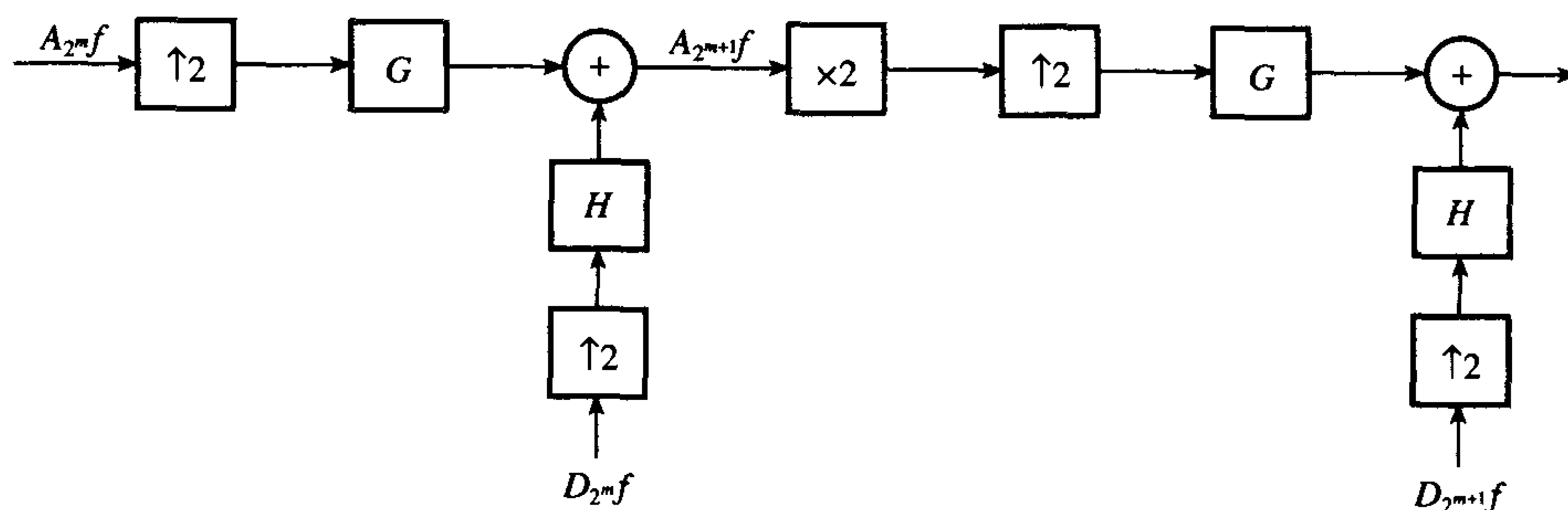
↓2 表示每隔一个抽样值保留一个

X 表示与标出的滤波器卷积

$A_x f$ 低通滤波信号，称为“粗”分量

$D_x f$ 高通滤波信号，称为信号“细节”，这些信号代表了两个邻近低通滤波信号之间信号的差别，即两个邻近分辨水平之间的差别

图3.10 应用小波变换方法的信号的MRA分解



↑2 表示每个抽样值之间插零

X 表示与标出的滤波器卷积

×2 表示乘2

$A_x f$ 低通滤波信号，称为“粗”分量

$D_x f$ 高通滤波信号，称为“细节”信号

图3.11 应用小波变换方法的信号的MRA重构

MRA 可以用来研究信号分量，也可以对信号滤波。信号分解后，不想要的分量就可以消除，再对滤波以后的信号进行重构。

图3.12给出了MRA在生物医学应用的典型结果，即将MRA应用到从信噪比为-14 dB的单次试验记录的（于噪声背景中）脑电信号（EEG）中提取诱发电压（CNV）。由于目的是要得到ERP，因此，通过对某些可变的ERP建模并且将他们加到EEG中来模拟很多试验。为了演示方法，这是可能实现的。自适应MRA（Saatchi et al., 1997）和它的修正算法都将应用到单次试验数据中。可以看出，去噪后的CNV波形与真实的CNV是很不相像的。由于相对大的ERP的信噪比大约是-15 dB，结果表明这些方法是很适合对具有相当大的信噪比的波形去噪。

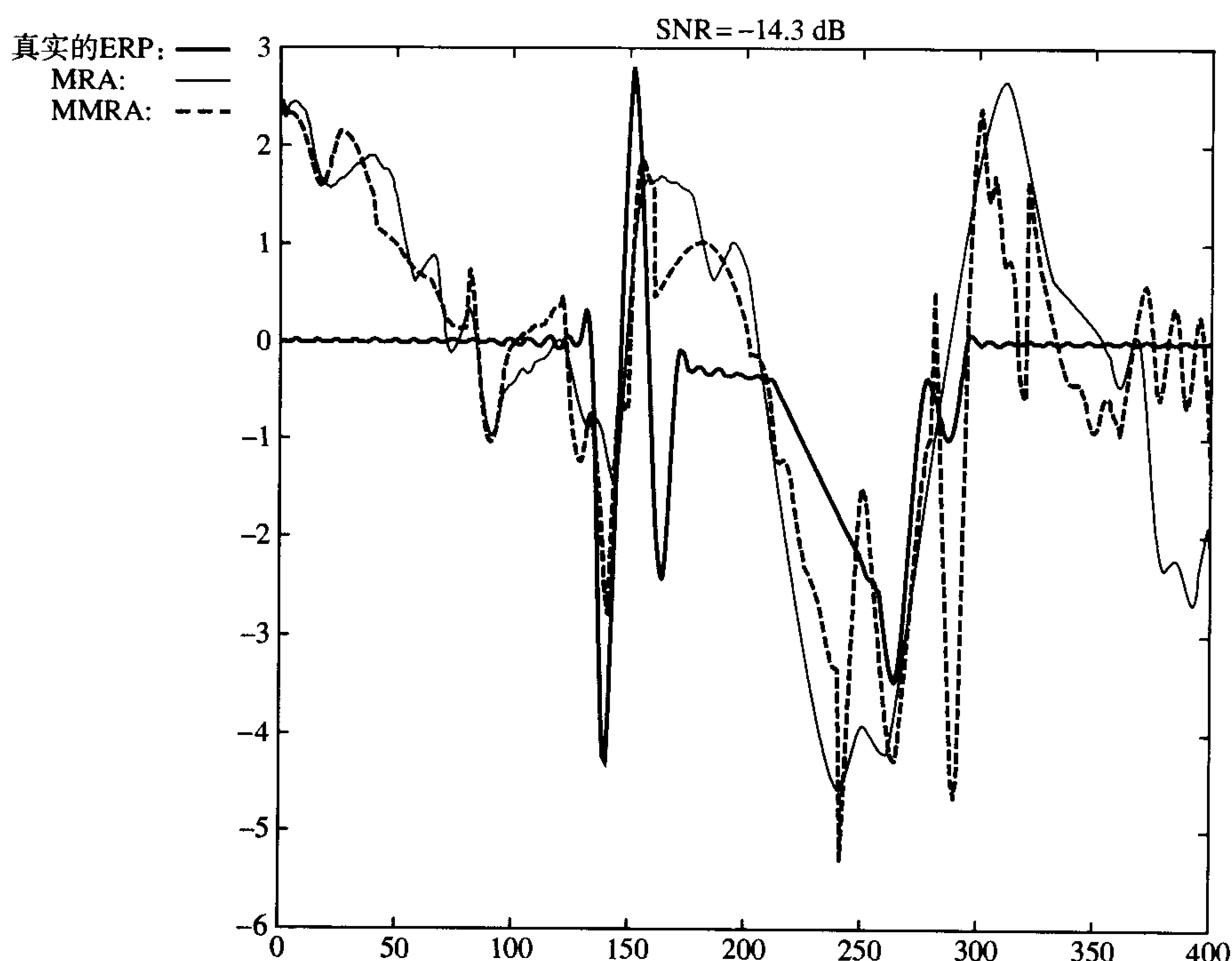


图 3.12 在信噪比为-14.3 dB时，应用多分辨分析从单次试验中提取的真实的和去噪后的ERP

3.8.6 信号的奇（点）异值表示：小波变换方法

已经证明（Mallat and Hwang, 1992），所有信号和噪声都可以完全由它们的奇异值表示。奇异值是根据Lipschitz指数定义的，如果某个函数$f(t)$在t_0处是连续可微的，那么它是非奇异的，我们说它的Lipschitz指数为1，Lipschitz指数不为1的信号就是奇异的。如果$f(t)$在t_0处是n次可微的，它的第n阶导数是奇异的，那么$f(t)$被描述成具有Lipschitz指数α，其中$\alpha>n$。具有负的Lipschitz指数也是可能的。Lipschitz指数用它的幅度和符号刻画了奇异值，根据奇异值来描述信号，然后消去不要的奇异值，滤波后的信号根据剩余的奇异值能够被重构。这种技术已经得到了应用，比如图像的边缘检测（Mallat and Hwang, 1992）和信号去噪（Mallat and Hwang, 1992; Zhang and Zheng, 1997）。信号去噪就是从信号中消去噪声，以提高信噪比。

许多信号都有正的Lipschitz指数的奇异值，而噪声是由负的Lipschitz指数来刻画的。因此，如果能够检测到与噪声有关的奇异值并且将其消去，我们就有可能从噪声中分离出信号。Mallat and Hwang(1992)已经证明能够找出奇异值，并且它由在不同尺度上的小波变换最大值与时间图上的模

和符号来刻画，如图3.13所示。因此，画出的每个最大值表示一个奇异值。通过比较不同尺度之间的最大值，就可能识别出与噪声有关的奇异值，并确定出Lipschitz指数的值。

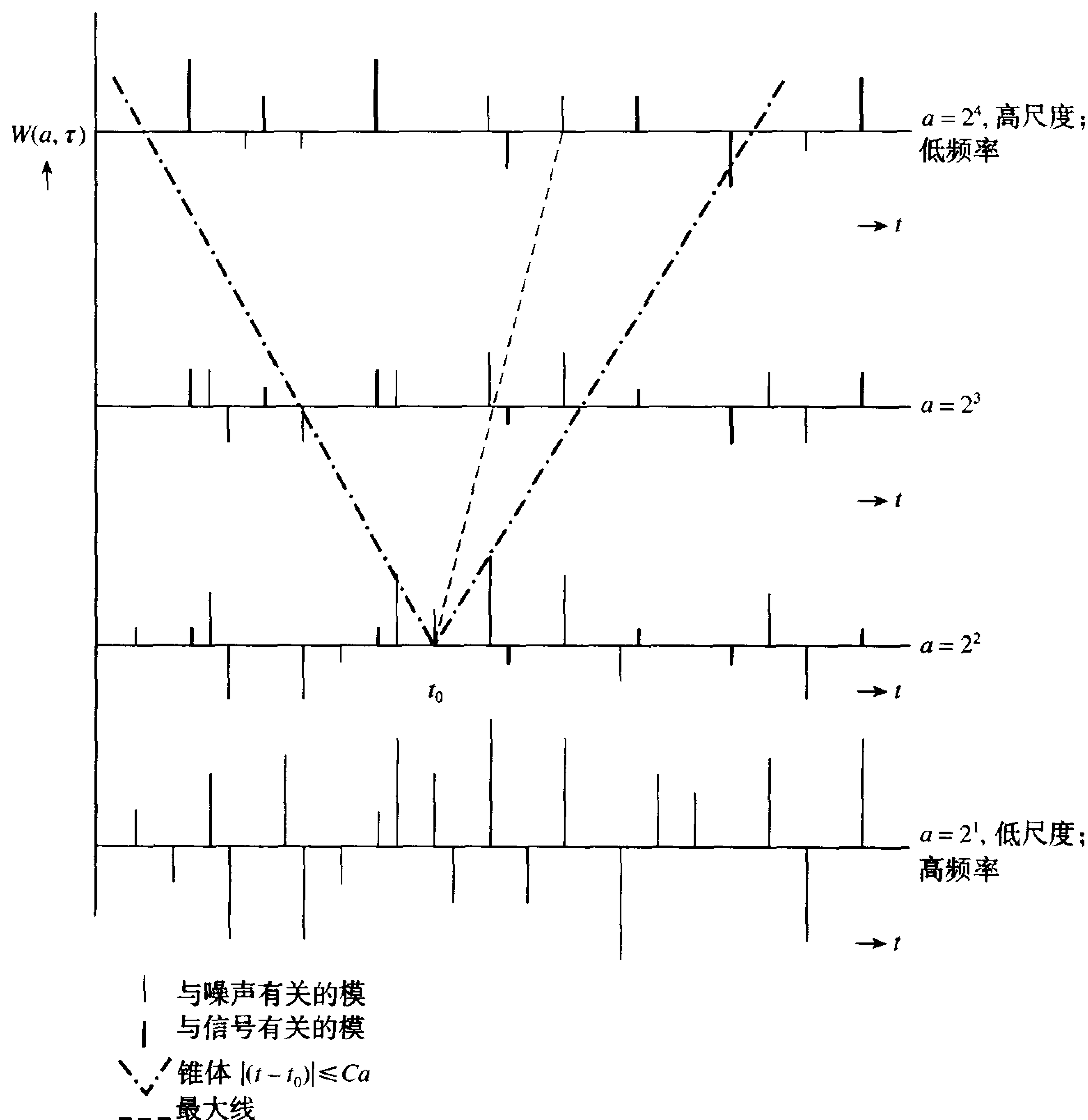


图3.13 对于不同的尺度因子a，小波变换最大值$W(a, \tau)$与时间图

在下文中我们假定信号不具有局部振荡特性，考虑局部振荡的信号要复杂得多（Mallat and Hwang, 1992）。

首先考虑小波变换模最大值位于锥体$|(t-t_0)| \leqslant Ca$内的信号，其中C是任意常数，这个锥体如图3.13所示。现在考虑锥体内的这些最大值，这些最大值位于不同尺度之间连接的曲线上，这条曲线称为最大线。小波变换$W(a, \tau)$与这条最大线上最大值的模有关，它是随尺度a变化的，即

$$|W(a, \tau)| \leqslant Aa^{\alpha} \tag{3.88}$$

其中A是一个常数，$a>0$和$0<\alpha<1$。

因此，随着a的增加，小波变换的模增加，也就是当变向低频尺度的时候，模会增加。所以，在低频段他们的信号能量内容增加，或者等价于高频段信号能量内容的减少。对3.88式取对数得到3.89式：

$$\log|W(a, \tau)| \leqslant \log A + \alpha \log a \tag{3.89}$$

这意味着Lipschitz指数是3.89式按对数标定的保持在$\log|W(a, \tau)|$上直线的最大斜率。后面我们会看到这是很有用的。

比较以上的情况，白噪声由负的Lipschitz指数来刻画，小波变换最大值模的平方的数学期望随下式变化，

$$E[|W(a,\tau)|^2]=\|\Psi\|^2\sigma^2/a \tag{3.90}$$

其中σ^2是噪声方差，Ψ是小波函数。因此，模的最大值随尺度增加按$1/\sqrt{a}$成比例减小。或者模最大值随频率按$\sqrt{a}$比例增加。这意味着与白噪声有关的最大值在整个尺度上随频率增加。这正好与具有正的Lipschitz指数的信号有关的幅度变化相反，对于其他像噪声的信号也是如此。

噪声信号的模最大值如图3.13所示，这只是图解，没有画出刻度。可以看出，在给定位置的信号最大值在高尺度时（低频）幅度增加，而在低尺度时（高频）噪声最大值幅度增加，并且最大值的数目也增加。一般来说，当尺度降低一半，噪声最大值的数目翻倍。因此，通过观察从尺度到尺度在同一位置最大值的变化，找出与噪声有关的这些最大值是可能的。因此，这些最大值可以消去，在去噪信号的重构中不再采用。

这项技术（Mallat and Hwang, 1992; Zhang and Zheng, 1997）已应用到3.8.5节中EEG数据的模拟ERP中。对于单次试验，去噪的和真实的ERP显示在图3.14中，其信噪比为-5.3 dB。相对于-15 dB来说这是大的信噪比，也是期望的最大信噪比，并且提取的CNV离真实的CNV差得远，为什么会这样呢？奇异值检测是基于白噪声假设，而EEG信号不是白色的。如果模拟用白噪声重复代替EEG，那么得出的结果如图3.15所示，这时，提取的CNV与真实的CNV非常类似。因此，要有满意的性能，必须为白噪声是很关键的。

当信号的最大值与噪声的最大值重合的时候会出现问题。如果最大值没有消去，那么噪声将保留在信号中。如果将最大值消去，那么某些信号也会被消去，使得重构的信号产生失真。解决这一问题要在信号占主要成分（相对大的信噪比）的地方寻找高的尺度，并且在锥体$|(t-t_0)|\leqslant Ca$内应用3.89式来确定α。锥体内相同位置在相邻低尺度的模最大值之比是2^α。因此，在这些低尺度处最大值的真实幅度可以计算出来，并且应用到信号的重构中。当然，对于任何模最大值，锥体$|(t-t_0)|\leqslant Ca$也可以按t_0画出。

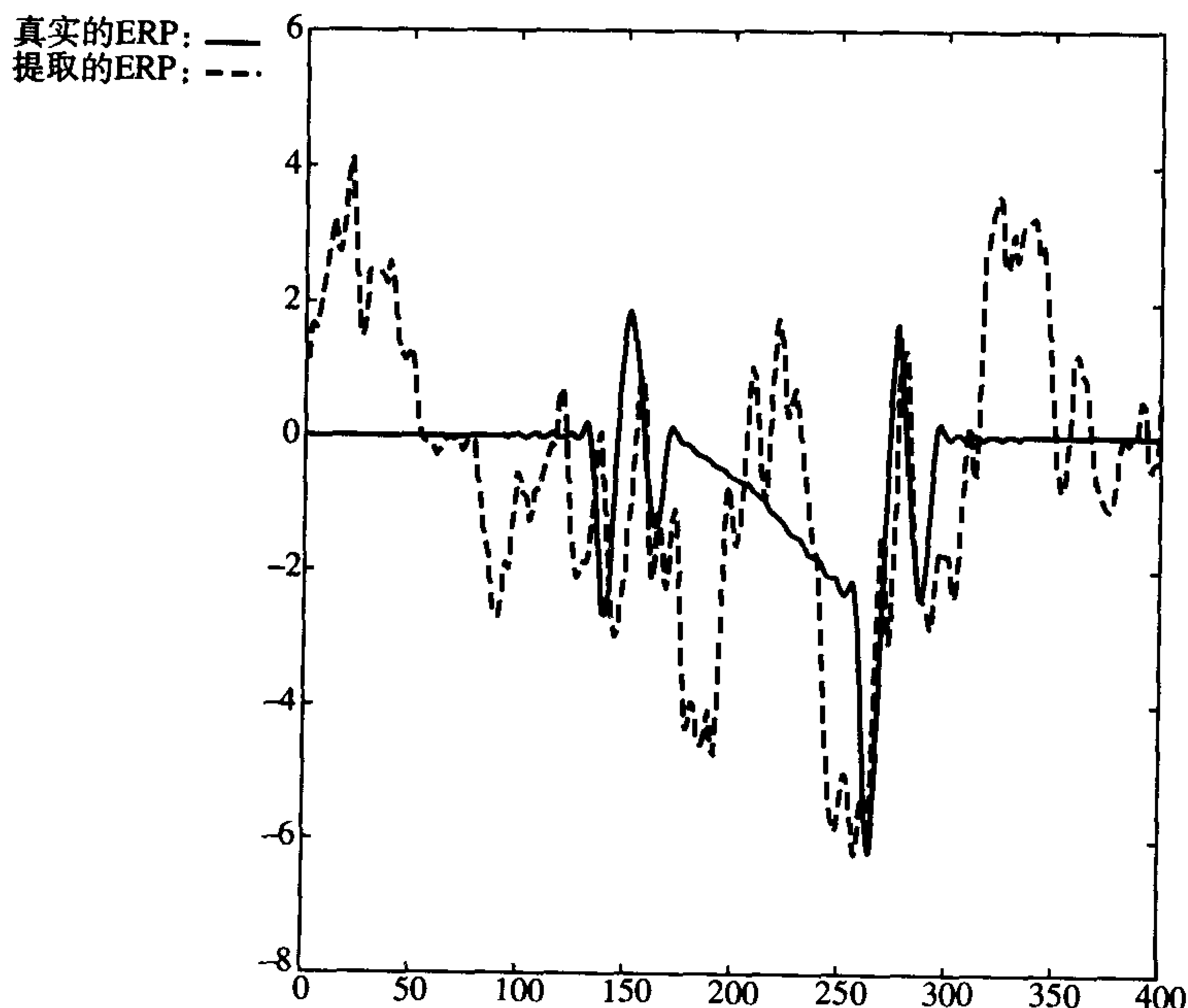


图3.14 用奇异值检测技术去噪的ERP和真实的ERP、EEG，信噪比为-5.3 dB

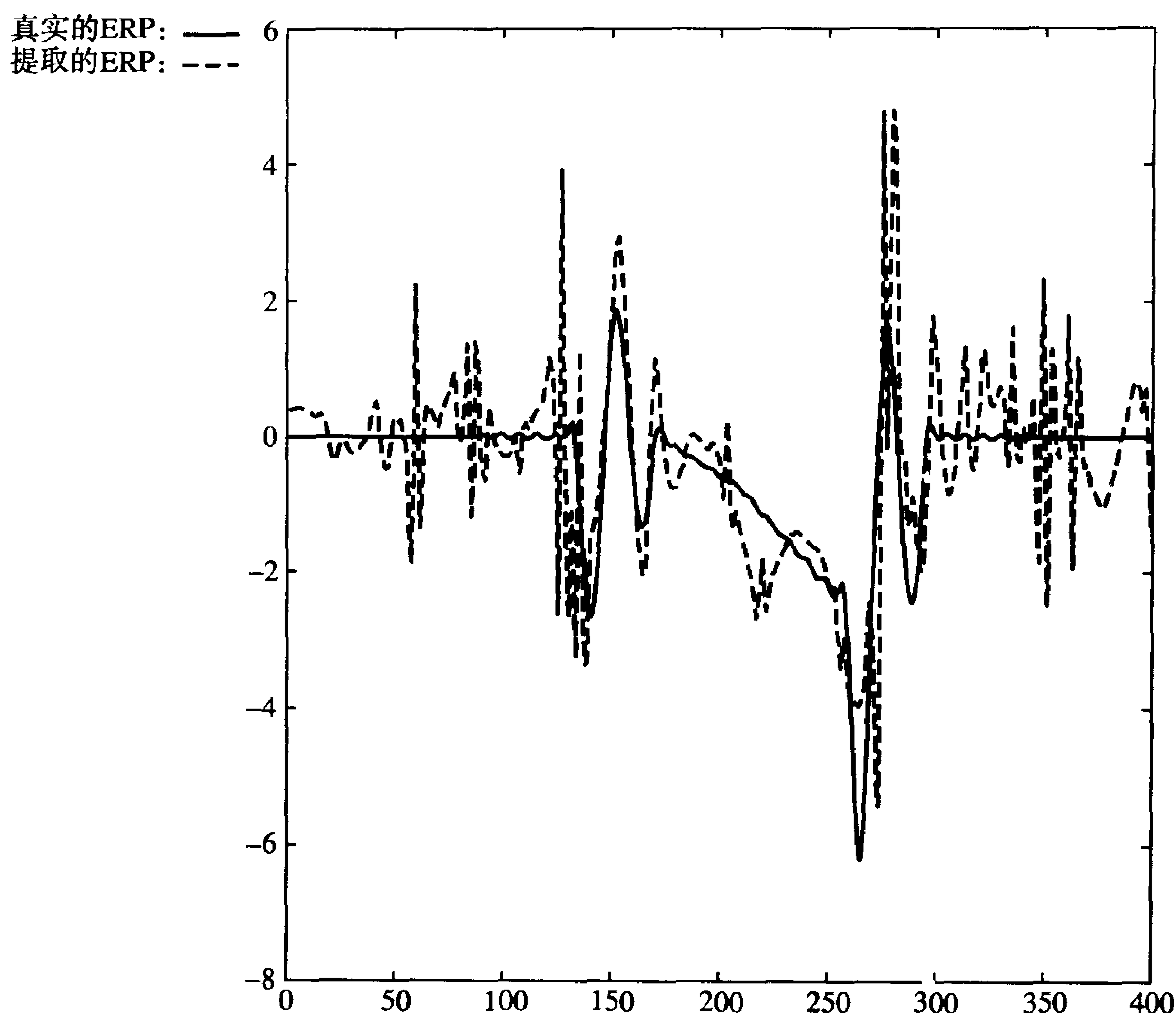


图 3.15 用奇异值检测技术去噪的 ERP 和真实的 ERP、白噪声，信噪比为 -5.67 dB

去噪技术的另一个限制是求噪声的模最大值并且消去的精度，这取决于计算α的精度，而计算α的精度又取决于选择的母小波以及奇异值（模最大值）消去算法和重构算法。相关细节读者可以参阅有关的参考文献。

3.9 DCT 的应用：图像压缩

正如3.8.1节所描述的那样，离散余弦变换（DCT）是用来压缩信号数据的，对于图像的存储和传输是特别重要的，因为每一幅图像都包含大量的数据。例如，由每像素8位表示的320 × 240图像元素（像素）将占76.8 kb，等价于25页文本。因此，图像压缩是十分重要的。在图像被压缩以后，为了传输通常都要进行编码，编码又导致进一步的压缩。不同的组织一直在开发自己的方法和标准。直到1986年，一群专家研究形成了联合图像专家组（Joint Photographic Experts Group, JPEG）的概念，试图将静止灰度和彩色图像的压缩和编码标准化。JPEG委员会是国际标准化组织的分委员会，这个委员会也包含了一些来自 CCITT（International Telegraph and Telephone Consultative Committee, 国际电话与电报顾问委员会）和 IEC（International Electrotechnical Commission，国际电工技术委员会）的成员。为了允许不同应用之间交换图像，如PC、LAN、CD-ROM、数码相机之间交换图像，标准是必需的。JPEG标准为一组图像压缩函数建立了一个结构，它允许在细节上有许多变化。有一种所有系统必须实现的基本结构，但是除了这一基本结构。还有扩展结构和分级结构。JPEG的成功鼓励了MPEG（Moving Pictures Expert Group，运动图像专家组）和JBIG（Joint Bi-level Image Experts Group，联合双态成像组）的形成。本节着重介绍JPEG标准的某些基本内容，在 Pennebaker and Mitchell(1993)中有完整的介绍。本书在附录中也引入了由 ISO DIS（Draft International Standard，国际标准草案）10918-1 发表的技术要求和指导方针，在Pitas(1993)、Bailey and Birch(1989)中有简单的描述。

图3.16是用于传输的JPEG压缩系统的基本框图，为了接收和解压缩需要一个逆系统。首先计算图像数据的二维DCT，然后DCT系数经过压缩和门限调整。对顺序零频或DC系数进行微分脉冲编码调制，得到的比特流是Huffman编码或者是算术编码。对其他频率系数（AC系数）进行Huffman编码或算术编码，对零的长游程进行游程长度（run-length）编码，结果产生两个压缩的数据流，这两个数据流是由编码的DC和AC系数组成。

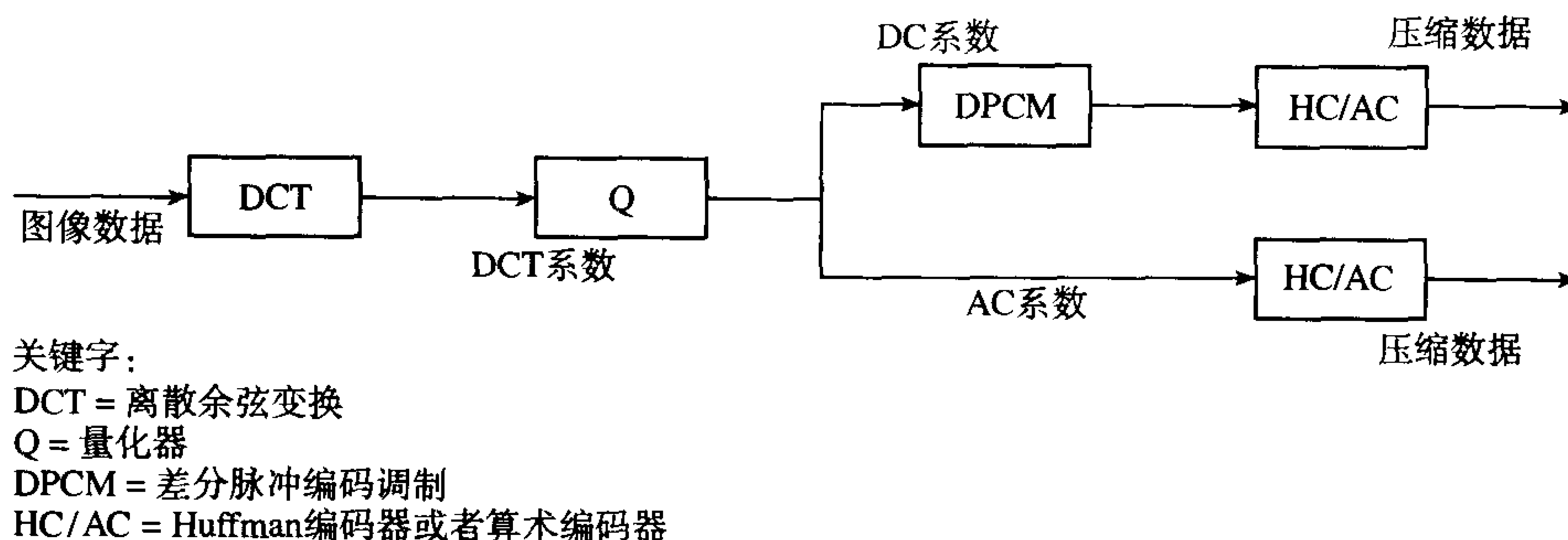


图3.16 JPEG数据压缩

3.9.1 离散余弦变换

任何矩形图像都可以用一个数值矩阵来表示，这个数值矩阵用某种方式指明了图像属性如亮度、色调和颜色，有关的详细内容请参见Pennebaker和Mitchell(1993)的著作。每一个值称为图像元素或像素，正如我们已经看到的那样，在一幅图像中有可能有很多像素。在不同的区域，图像的统计可能有很大的不同。因此，为了变换最好是把一幅图像分成许多连续的小块，这些小块有类似的统计特性。小块也可以产生更高的压缩，因为邻近像素之间相关性更强。此外，小块的变换也更容易计算。因此，在JPEG标准中，基本块是由8 × 8像素的方块组成，所以图像被细分为8 × 8像素块的紧接的邻块。

这些8 × 8像素块是二维（2D）的，适合采用二维DCT对块进行变换。这可以采用如下方法实现：首先计算像素的每一水平行的DCT，用DCT分量（水平DCT）取代像素的水平行；然后计算列的DCT，用它的DCT（垂直DCT）取代每一列。由于水平DCT的分量频率从左到右是增加的，垂直DCT的分量自上而下是增加的，得到的2D DCT在它的左上部分包含低的频率分量，而在右下部分包含高的频率分量。由于低频分量常常有比高频分量更大的幅度，因此在图的左上角趋向于包含相对大的值，而在右下角趋向于包含小的值。在JPEG标准中，2D DCT（Pennebaker and Mitchell, 1993）为

$$S(v,u)=\frac{1}{4}C(v)C(u)\sum_{x=0}^{7}s(y,x)\cos\{(2x+1)u\pi/16\}\cos\{(2y+1)v\pi/16\} \tag{3.91}$$

其中，$S(v,u)$是2D DCT系数，

$$C(v)=\begin{cases}1/\sqrt{2}, & v=0\\ 1\ , & v>0\end{cases}$$

$$C(u)=\begin{cases}1/\sqrt{2}, & u=0\\ 1\ , & u>0\end{cases}$$

$s(x,y)$是8 × 8像素块中的像素值。

图像重构要求的2D DCT反变换为

$$s(y,x)=\frac{1}{4}\sum_{u=0}^{7}\sum_{v=0}^{7}C(u)C(v)S(v,u)\cos\{(2x+1)u\pi/16\}\cos\{(2y+1)v\pi/16\} \tag{3.92}$$

和DCT一样，存在快速2D DCT变换（Pennebaker and Mitchell, 1993）。

3.9.2 2D DCT系数量化

64个DCT的每一个系数$S(v,u)$采用均匀量化器单独量化（参见2.4.2节），64个量化器中的每一个有不同的步长，每一个系数用它的量化器的量化步长归一化，其结果再舍入到最近的整数。这一过程产生一个整数矩阵，这个矩阵包含一些零，特别是在矩阵的右下角。

3.9.3 编码

矩阵中左上角的系数对应于2D DCT的DC项，它表示8 × 8块的平均信号电平，它的值通常在邻近的块之间是不会迅速变化的。因此，对这个系数的处理不同于其他AC系数（参见图3.16），它用差分脉冲编码调制（DPCM）进行差分编码。其值设定为它的值与来自于前面8 × 8块的DC系数值之差，通常会得到相对较小的值。

AC系数按照锯齿（zig-zag）序列的顺序依次排列，锯齿序列是取自8 × 8的矩阵，这个锯齿序列如图3.17所示，序列跟随着矩阵内的数，目的是对2D系数从最大到最小排序。

$$\begin{pmatrix}
0 & 1 & 5 & 6 & 14 & 15 & 27 & 28\\
2 & 4 & 7 & 13 & 16 & 26 & 29 & 42\\
3 & 8 & 12 & 17 & 25 & 30 & 41 & 43\\
9 & 11 & 18 & 24 & 31 & 40 & 44 & 53\\
10 & 19 & 23 & 32 & 39 & 45 & 52 & 54\\
20 & 22 & 33 & 38 & 46 & 51 & 55 & 60\\
21 & 34 & 37 & 47 & 50 & 56 & 59 & 61\\
35 & 36 & 48 & 49 & 57 & 58 & 62 & 63
\end{pmatrix}$$

图3.17 2D DCT锯齿序列

接着对两组系数进行编码（参见图3.16），从而进一步压缩数据。当更长的游程出现时，对它们就可以采用游程长度编码。对这些零进行压缩，这个码指明了连续有多少个零。当出现在锯齿的末端时，使用块码字结束标志。剩余的系数是在基线顺序系统中用Huffman编码，或者在扩展DCT方案中用算术编码。两种码都含有可变长度码字，最常出现的码字是最短的，这减少了发送的位数，换句话说是增加了压缩度。这两种码以最少的位数发送最多的信息，从这个意义上说它们是最有效的。Huffman码使用了最佳选择的码字集，它包含了信息位的整数。采用算术编码大约增加10%的压缩性能，它是一种单次通过（one-pass）的自适应编码的形式，采用这种编码形式的码本动态地自适应正在编码的数据。然而，有关信息论与编码的主题超出了本书的范围。

3.10 处理过的例子

例3.8 10 Hz带宽的信号以125 Hz抽样，前四个抽样值为(0.5, 1, 1, 0.5)，举例说明如何通过快速傅里叶变换得到这个序列的离散傅里叶变换，因而也就得到了数据的傅里叶变换。

解：

FFT的流图如3.18所示，我们有

$$X(0)=G(0)+W_4^0H(0)=G(0)+H(0)$$

$$G(0)=x(0)+W_2^0x(2)=x(0)+x(2)$$

$$H(0)=x(1)+W_2^0x(3)=x(1)+x(3)$$

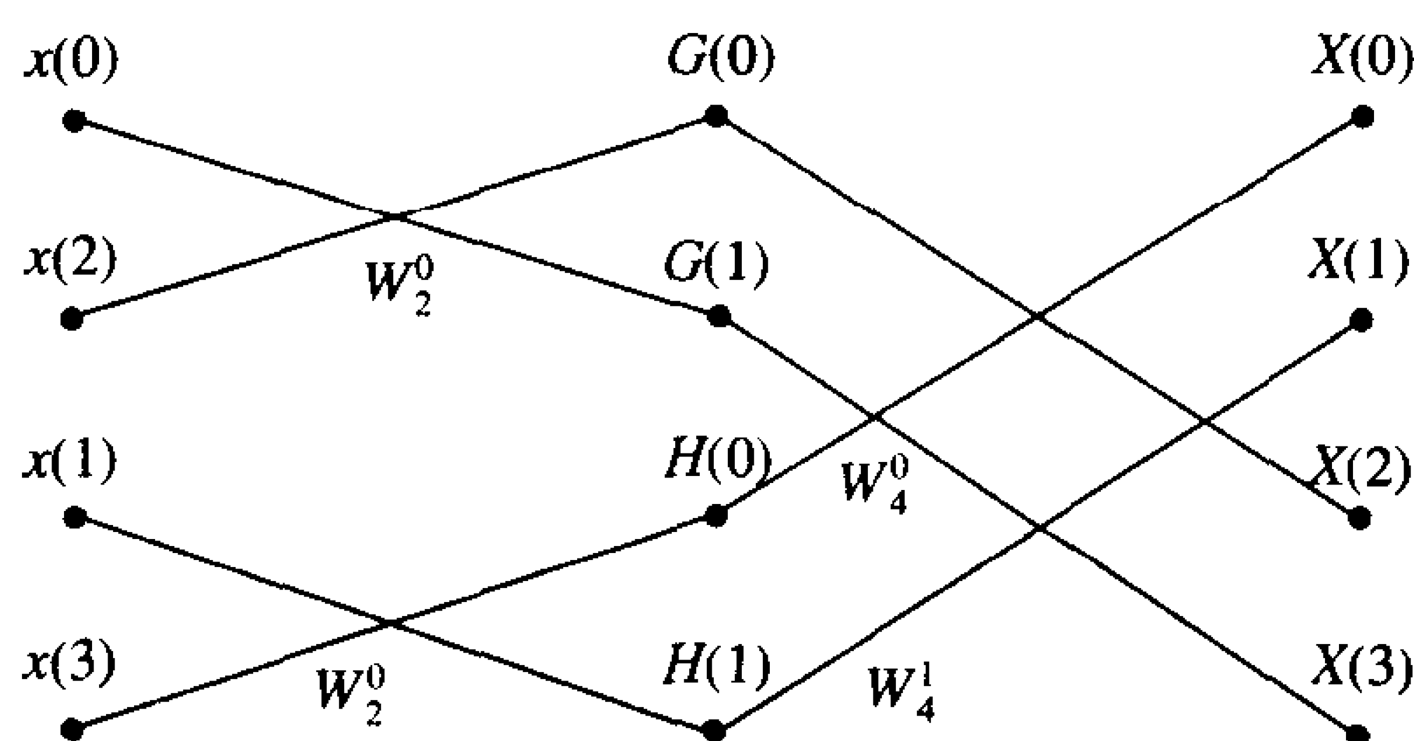


图 3.18 例 3.8 中 FFT 的流图

由于 $W^0=1$，代入得

$$X(0)=x(0)+x(2)+x(1)+x(3)$$
$$=0.5+1+1+0.5=3$$
$$X(1)=G(1)+W_4^1H(1)$$
$$G(1)=x(0)-W_2^0x(2)=x(0)-x(2)$$
$$H(1)=x(1)-W_2^0x(3)=x(1)-x(3)$$

而 $W_N=\mathrm{e}^{-\mathrm{j}2\pi/N}$，因此$W_4^1=\mathrm{e}^{-\mathrm{j}2\pi/4}=\mathrm{e}^{-\mathrm{j}\pi/2}$，代入得

$$X(1)=x(0)-x(2)+\mathrm{e}^{-\mathrm{j}\pi/2}[x(1)-x(3)]$$
$$=0.5-1+\left[\cos\left(\frac{\pi}{2}\right)-\mathrm{j}\sin\left(\frac{\pi}{2}\right)\right](1-0.5)$$
$$=0.5-1+(0-\mathrm{j})0.5=-0.5-\mathrm{j}0.5=-0.5(1+\mathrm{j})$$
$$X(2)=G(0)-W_4^0H(0)=G(0)-H(0)$$
$$=x(0)+x(2)-[x(1)+x(3)]$$
$$=0.5+1-(1+0.5)=0$$
$$X(3)=G(1)-W_4^1H(1)$$
$$=x(0)-x(2)-\mathrm{e}^{-\mathrm{j}\pi/2}[x(1)-x(3)]$$
$$=0.5-1-\left[\cos\left(\frac{\pi}{2}\right)-\mathrm{j}\sin\left(\frac{\pi}{2}\right)\right](1-0.5)$$
$$=-0.5-(-\mathrm{j})0.5=-0.5+\mathrm{j}0.5=0.5(-1+\mathrm{j})$$

所以

$$X(\Omega)=\{3,-0.5(1+\mathrm{j}),0,0.5(-1+\mathrm{j})\}$$

其中抽样间隔 T 取小值，

$$\mathrm{FT}=T\,\mathrm{DFT}$$

其中 FT 是傅里叶变换。这里，$T=1/125$ 秒 $=0.008$ 秒，信号周期是 1/10 秒 $=0.1$ 秒，所以，

$$\frac{T}{\text{周期}}=\frac{0.008}{0.1}=0.08\ll 1$$

这是 FT = T DFT 的一个好的近似，所以，

$$\mathrm{FT}=\{0.024,-0.004(1+\mathrm{j}),0,0.004(-1+\mathrm{j})\}$$

例3.9 在数据压缩系统中，数据首先被变换，然后对变换的值进行门限调整，设定的门限为0.375。考虑两个变换——离散余弦变换和沃尔什变换，$X_c(K)$离散余弦变换定义为

$$X_c(k) = \frac{1}{N}\sum_{n=0}^{N-1} x_n \cos\left(\frac{k2\pi n}{N}\right), \quad k = 0, 1, \ldots, N-1$$

沃尔什变换定义为

$$X_k = \frac{1}{N}\sum_{i=0}^{N-1} x_i \,\mathrm{WAL}(k, i), \quad k = 0, 1, \ldots, N-1$$

假定数据序列为{1, 2, 0, 3}，求

(1) 在这种情况下，对于数据压缩，哪种变换更有效；

(2) 达到的数据压缩百分比。

对用沃尔什变换得到的压缩数据求反变换，并与原始数据序列进行比较。

解：

(1) 用 $x_0 = 1$、$x_1 = 2$、$x_2 = 0$、$x_3 = 3$ 计算DCT，

$$X_c(0) = \frac{1}{4}(x_0 \cos 0 + x_1 \cos 0 + x_2 \cos 0 + x_3 \cos 0)$$

$$= \frac{1}{4}(1 + 2 + 0 + 3) = \frac{6}{4} = 1.5$$

$$X_c(1) = \frac{1}{4}\sum_{n=0}^{3} x_n \cos\left(\frac{2\pi n}{4}\right) = \frac{1}{4}\sum_{n=0}^{3} x_n \cos\left(\frac{n\pi}{2}\right)$$

$$= \frac{1}{4}\left[x_0 + x_1 \cos\left(\frac{\pi}{2}\right) + x_2 \cos\left(\frac{2\pi}{2}\right) + x_3 \cos\left(\frac{3\pi}{2}\right)\right]$$

$$= \frac{1}{4}[1 + 2 \times 0 + 0 \times (-1) + 3 \times 0] = 0.25$$

$$X_c(2) = \frac{1}{4}\left[x_0 \cos\left(\frac{4\pi \times 0}{4}\right) + x_1 \cos\left(\frac{4\pi \times 1}{4}\right) + x_2 \cos\left(\frac{4\pi \times 2}{4}\right) + x_3 \cos\left(\frac{4\pi \times 3}{4}\right)\right]$$

$$= \frac{1}{4}(x_0 - x_1 + x_2 - x_3) = \frac{1}{4}(1 - 2 + 0 - 3) = -1$$

$$X_c(3) = \frac{1}{4}\left[x_0 \cos\left(\frac{6\pi \times 0}{4}\right) + x_1 \cos\left(\frac{6\pi}{4}\right) + x_2 \cos\left(\frac{6\pi \times 2}{4}\right) + x_3 \cos\left(\frac{6\pi \times 3}{4}\right)\right]$$

$$= \frac{1}{4}[1 + 2 \times 0 + 0 \times (-1) + 3 \times 0] = 0.25$$

所以

$$\mathrm{DCT} = \{1.5, 0.25, -1, 0.25\}$$

经过门限调节（|值|>0.375）后剩余的值是1.5和−1。

对于沃尔什变换，在3.8.2节已经计算出 $X_k = \{1.5, 0, 0.5, -1\}$，所以经门限调整后剩余的值是1.5、0.5和−1。因此，在这种情况下，DCT提供了更有效的数据压缩。

(2) 假定数据压缩效率为 η，它的定义为

$$\eta = \frac{（原始序列中的数据个数 - 变换后序列数据的个数）\times 100\%}{原始序列中的数据个数}$$

那么，

$$\eta = \frac{4-2}{4} \times 100\% = 50\%$$

最后，经沃尔什变换压缩后的数据是{1.5, 0, 0.5, −1}，由 3.70 式给出的反变换为

$$x_i = \sum_{i=0}^{N-1} X_k \ \mathrm{WAL}(k, i), \quad i = 0, 1, \ldots, N-1$$

$$x_i = [1.5 \quad 0 \quad 0.5 \quad -1]\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} = [1 \quad 2 \quad 0 \quad 3]$$

这与原始数据是相同的，这是因为 $X_1 = 0 < 0.375$，尽管没有将其发送。在反变换中用 0 表示，这是它的精确值。通常，重构序列是原始序列的近似。

习题

3.1 求图 3.19 中周期波形的傅里叶级数表示。

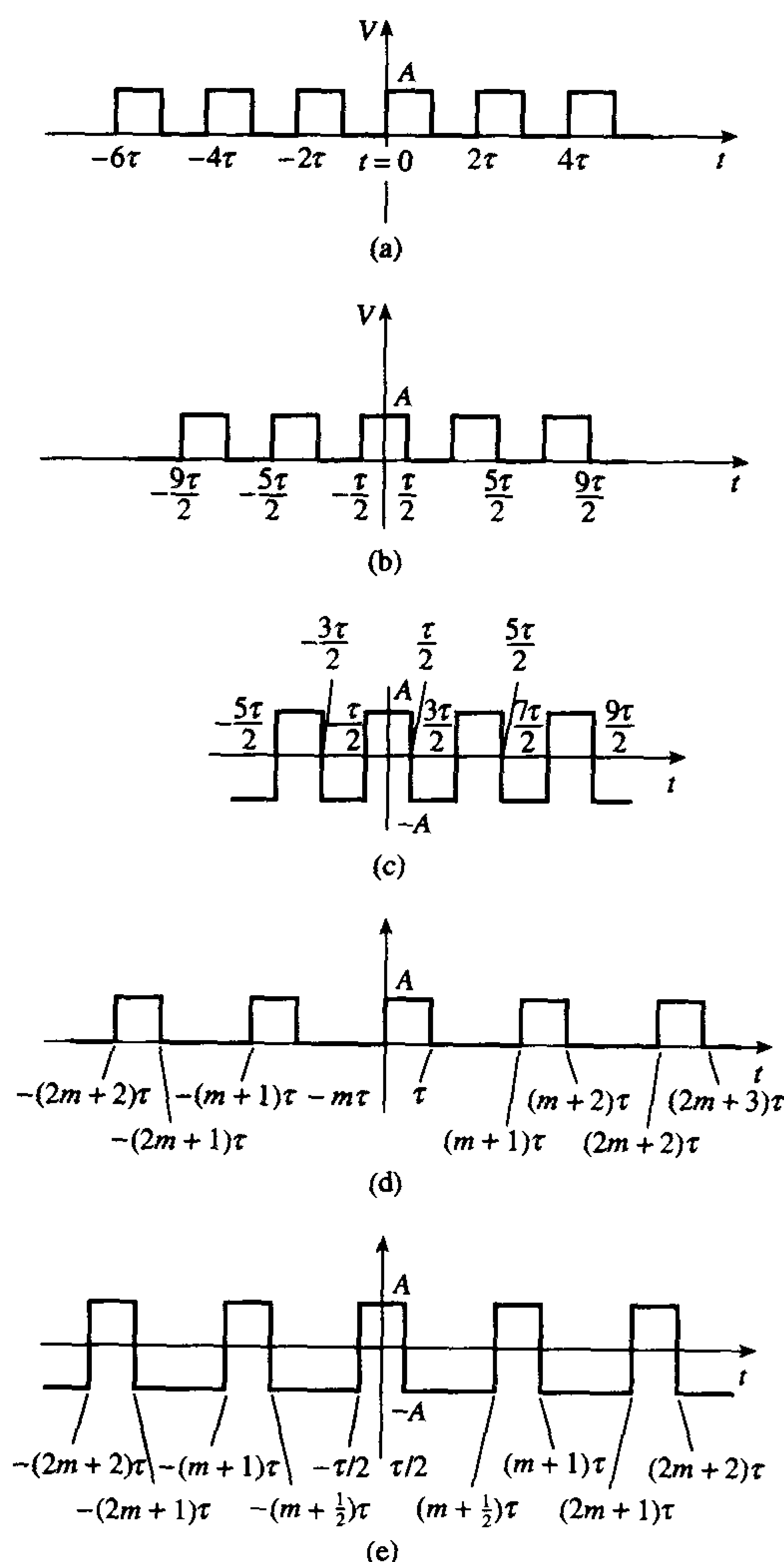


图 3.19 习题 3.1 ~ 习题 3.4 的周期波形

3.2　求图 3.19 中波形的复傅里叶级数表示。

3.3　证明习题 3.1 中求得的幅度与习题 3.2 中求得的幅度一致。

3.4　画出图 3.19 中波形的傅里叶分量的幅度谱和相位谱。

3.5　计算由下式给出的电压波形 $v(t)$的幅度和能量谱密度，

$$v(t)=\begin{cases}\dfrac{A}{\tau}t+A & -\tau\leqslant t\leqslant 0\\ -\dfrac{A}{\tau}t+A & 0\leqslant t\leqslant \tau\\ 0 & \tau\leqslant t\leqslant -\tau\end{cases}$$

其中 $A=5\ \mathrm{V}$，$\tau=20\ \mathrm{ms}$。

3.6　计算由下式给出的波形 $v(t)$的能量谱密度和相位谱，

$$v=\begin{cases}2\sin\left[\dfrac{2\pi}{T}\left(t+\dfrac{T}{4}\right)\right] & -\dfrac{T}{4}\leqslant t\leqslant \dfrac{T}{4}\\ 0 & \text{其他}\end{cases}$$

其中 $T=0.0167\ \mathrm{s}$。

3.7　画出下列函数的能量谱密度，

$$w(t)=\begin{cases}\sin\left[\dfrac{2\pi}{T}\left(t+\dfrac{3T}{4}\right)\right] & -\dfrac{3T}{4}\leqslant t\leqslant -\dfrac{T}{2}\\ 1.0 & -\dfrac{T}{2}\leqslant t\leqslant \dfrac{T}{2}\\ \sin\left[\dfrac{2\pi}{T}\left(t-\dfrac{3T}{4}\right)\right] & \dfrac{T}{2}\leqslant t\leqslant \dfrac{3T}{4}\\ 0 & \dfrac{3T}{4}\leqslant t\leqslant -\dfrac{3T}{4}\end{cases}$$

其中 $T=4$ 秒。

3.8　计算数据序列{0, 1, 1, 0}的 DFT，通过计算它的 IDFT 检查答案的正确性。

3.9　推导 $X(k)$和 $X^2(k)$的量纲，然后计算并画出数据序列{0, 1, 1, 0}的能量谱，它的 DFT 是习题 3.8 计算的结果。

3.10　如果习题 3.8 的序列{0, 1, 1, 0}表示以 125 Hz 抽样所得到的数字化样本，求数据序列的傅里叶变换的能量谱密度和相位谱。

3.11　利用DFT的时移特性和习题3.8的解求时间序列{0, 0, 0, 0, 0, 1, 1, 0}的幅度和相位谱，时间序列是以在时刻 $t=0, 1, 2,\ldots, 7$ ms 抽样得到的数据。

3.12　用习题 3.9 的结果对数据{0, 1, 1, 0}验证 Parseval 定理。

3.13　用相关定理计算数据序列{1, 1, 0, 1}和{1, 0, 0, 1}的循环相关，画出相关与延迟数 j 的关系图。

3.14　用相关定理计算数据序列{1, 1, 0, 1}和{1, 0, 0, 1}的线性相关，画出相关函数与延迟数的关系图，并与习题 3.13 的解进行比较，对任何差别进行解释。

3.15　用时域抽取（Cooley-Tukey）FFT 算法计算数据序列{0, 1, 1, 0}的 DFT，将结果与习题 3.8 的结果进行核对，比较两种方法中复数加法和复数乘法的次数。

3.16　求习题 3.15 结果的 IFFT，验证求得的数据序列为{0, 1, 1, 0}。

3.17 计算数据序列{0, 0, 1, 1, 1, 1, 0, 0}的FFT，画出幅度和相位谱，通过计算它的IFFT得到原始数据序列来核对你的结果。

3.18 编写计算FFT和IFFT的计算机程序，通过计算习题3.8的数据序列{0, 1, 1, 0}、习题3.13的数据序列{1, 1, 0, 1}和{1, 0, 0, 1}的DFT来检查FFT程序，通过计算DFT序列的IDFT来检查IFFT程序。

3.19 用FFT计算习题3.5和习题3.7的波形的1024点DFT，画出它们的能量和相位谱，并与习题3.5和习题3.7得到的图进行比较。

3.20 用1024点FFT计算并画出幅度为5 V、宽度为$\tau = 6$秒的矩形脉冲的能量谱，并与习题3.7的结果进行比较。

3.21 (1) 用卷积定理（3.37式）求下面两对波形的谱的卷积：

(a) $v_s = \sin(2\pi \times 100t)$和中心在$t = 0$、宽度为2秒的单位高度脉冲$v_w$；

(b) $v_s = \sin(2\pi \times 100t)$和

$v_w = \cos(2\pi \times 0.25t) \quad 1 \leqslant t \leqslant -1$ 秒

$\quad 0 \quad$ 其他

(2) 当信号的傅里叶分量是用抽样数据的DFT得出的时候，信号抽样值有效使用的长度是$(N-1)T$，其中N是数据个数，T是抽样间隔。我们说用长度$(N-1)T$的窗口对数据加窗，那么计算的谱由信号的谱和窗函数的谱卷积得到。在(1)项的情况中，v_s表示信号，v_w表示窗数据。解释两个数据窗口对所定义的信号抽样值的适应性。

3.22 计算数据序列{0.1, −0.2, 0.3, −0.4, 0.5, 1.5, 2, 1.5, 0.5, −0.4, 0.3, −0.2, 0.1}的离散余弦、离散沃尔什和离散哈达玛变换。比较这些变换的压缩效率，预选的门限值是0.35，按照优先顺序进行排序。

3.23 通过扫描照片图像的强度分布得到的抽样电压是{3.2, 3.6, 3.3, 2.9, 1.7, 1.6, 1.8, 1.5}，讨论用FFT和DWT变换这些数据的相对优点。

3.24 扩展习题3.23的讨论，包括有效的数据压缩量（包括使用DCT）。

3.25 画一个表来说明应用快速傅里叶变换、离散沃尔什变换和离散哈达玛变换的优缺点。

MATLAB 习题

3.26 (a) 利用直接法采用合适的MATLAB函数求下列8点离散时间序列的DFT系数：

$$x(n) = \{4, 2, 1, 4, 6, 3, 5, 2\}$$

(b) 用合适的MATLAB函数求对应于下列DFT系数的离散时间序列：

27 + 0j
−4.121 32 + 3.292 893j
4 + j
0.121 32 − 4.707 107j
5 + 0j
0.121 32 + 4.707 107j
4 − j
−4.121 32 − 3.292 893j

3.27 (a) 用MATLAB计算由下式给出的离散时间序列的32点FFT：

$$x(n) = \begin{cases} 1, & n = 0, 1, \ldots, 15 \\ 0, & n = 16, 17, \ldots, 31 \end{cases}$$

(b) 用MATLAB计算(a)中的数据序列的64点FFT。

(c) 比较(a)和(b)得到的结果。

3.28 解释基-2FFT算法是如何求离散时间系统频率响应的估计，系统的 z 传递函数具有有理多项式的形式。通过如下例子说明你的结果：用MATLAB的FFT函数求具有下列 z 传递函数的离散时间滤波器的频率响应，

$$H(z)=\frac{1-1.618z^{-1}+z^{-2}}{1-1.516z^{-1}+0.87z^{-2}}$$

指出实际中可能涉及到的问题。

参考文献

Ahmed N. and Rao K.R. (1975) *Orthogonal Transforms for Digital Signal Processing*. Berlin: Springer.

Bailey D.J. and Birch N. (1989) Image compression using a discrete cosine transform image processor. *Electronic Engineering*, July, 9–44.

Beauchamp K.G. (1987) *Transforms for Engineers. A Guide to Signal Processing*. Oxford: Clarendon.

Burrus C.S. (1998) *Introduction to Wavelets and Wavelet Transforms: A Primer*. Englewood Cliffs NJ: Prentice-Hall.

Burrus C.S. and Parks T.W. (1985) *DFT/FFT and Convolution Algorithms. Theory and Implementation*. New York: Wiley.

Chan Y.T. (1995) *Wavelet Basics*. Boston MA: Kluwer Academic.

Chen W., Smith C.H. and Fialick S.C. (1977) A fast computational algorithm for the discrete cosine transform. *IEEE Trans. Communications*, **25**, 1004–9.

Chui C.K. (1992) *An Introduction to Wavelets*. Boston MA: Academic Press.

Cooley J.W. and Tukey J.W. (1965) An algorithm for the machine calculation of complex Fourier series. *Mathematics Computation*, **19**, 297–301.

Daubechies I. (1990) The wavelet transform, time-frequency localisation and signal analysis. *IEEE Trans. Information Theory*, **36**(5), 961–1005.

Daubechies I. (1992) *Ten Lectures on Wavelets*. Philadelphia: The Society for Industrial and Applied Mathematics.

Gentleman W.M. and Sande G. (1966) Fast Fourier transforms for fun and profit. In *Fall Joint Computing Conf., AFIPS Proc.*, **29**, 563–78.

Mallat S.G. (1989) A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **11**(7), 674–93.

Mallat S. and Hwang W.L. (1992) Singularity detection and processing with wavelets. *IEEE Trans. Information Theory*, **38**(2), 617–43.

McClellan J.H. and Rader C.M. (1979) *Number Theory in Digital Signal Processing*. Englewood Cliffs NJ: Prentice-Hall.

Narasinka M.J. and Petersen A.M. (1978) On the computation of the discrete cosine transform. *IEEE Trans. Communications*, **26**, 934–6.

Pennebaker W.B. and Mitchell J.L. (1993) *JPEG Still Image Data Compression Standard*. New York: Van Nostrand Reinhold.

Pitas I. (1993) *Digital Image Processing Algorithms*. New York: Prentice-Hall.

Rader C.M. (1968) Discrete Fourier transform when the number of data samples is prime. *IEEE Proc.*, **56**, 1107–8.

Rosenfield A. and Thurston M. (1971) Edge and curve detection for visual scene analysis. *IEEE Trans. Computing*, **20**, 562–9.

Saatchi M.R., Gibson C. and Rowe J.K.W. (1997) Adaptive multiresolution analysis based evoked potential filtering. *IEE Proc.-Sci. Meas. Technol.*, **144**(4), July, 149–55.

Signal Processing Committee (ed.) (1979) *Programs for Digital Signal Processing*. New York: IEEE.

Srinivassan R. and Rao K.R. (1983) An approximation to the discrete cosine transform. *Signal Processing*, **5**, 81–5.

Strum R.D. and Kirk D.E. (1988) *First Principles of Discrete Systems and Digital Signal Processing*. Reading MA: Addison-Wesley.

Thakor N.V., Xin-Rong G., Yi-Chun S. and Hanley D.F. (1993) Multiresolution wavelet analysis of evoked potentials. *IEEE Trans. Biomedical Engineering*, **40**(11), 1085–93.

Winograd S. (1978) On computing the discrete Fourier transform. *Mathematics Computation*, **32**, 175–99.

Yip P. and Ramamohan K. (1987) In *Handbook of Digital Signal Processing Engineering Applications* (Elliott D.E. (ed.)). New York: Academic Press.

Zhang J. and Zheng C. (1997) Extracting evoked potentials with the singularity detection technique. *IEEE Engineering in Medicine and Biology*. 155–61.

附录

3A 离散DFT计算的C语言程序

这里给出的C语言程序直接计算离散时间序列$x(n)$的DFT或IDFT：

$$X(k) = \sum_{n=0}^{N-1} x(n)W^{nk}, \quad k = 0, 1, \ldots, N-1 \qquad \text{DFT} \qquad (3A.1a)$$

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)W^{-nk} \qquad \text{IDFT} \qquad (3A.1b)$$

其中，$W = e^{-j2\pi/N}$，N是序列的长度。

输入序列$x(n)$必须是复数形式（实部和虚部）。对于实数据序列，数据的虚部设为零。程序3A.1列出了主函数DFTD.c的清单，程序3A.2列出了计算DFT或IDFT的函数清单。两个函数read_data()和save_data()用来读输入数据序列和保存变换的数据（参见程序3A.3）。输入数据放在输入文件coeff.dat中，输出数据保存在文件dftout.dat中。

程序3A.1 主函数dftd.c，驱动DFT的计算

```
/*-----------------------------------------------------------------------------------------*/
/*                                                                                         */
/*          Program to compute DFT coefficients directly                                   */
/*          3 other functions are used                                                     */
/*                                                                                         */
/*          E C Ifeachor. July, 1992                                                       */
/*                                                                                         */
/*-----------------------------------------------------------------------------------------*/
#include    "dsp1.h"
#include    "dft.h"

main()
{
            extern  long npt;
            extern  int     inv;

            printf("select type of transform\n");
            printf("\n");
            printf("0      for forward DFT\n");
            printf("1      for inverse DFT\n");
            scanf("%d", &inv);
            read_data();
            dft();
            save_data();
            exit();
}
#include    "dft.c";
#include    "rdata.c";
#include    "sdata.c";
```

程序3A.2 直接计算离散时间序列DFT的C语言函数，这一函数单独放在一个文件中

```
/*-----------------------------------------------------------------------------------------*/
/*                                                                                         */
/*          Function to compute the DFT of a discrete-time                                 */
/*          sequence directly                                                              */
/*                                                                                         */
/*          E C Ifeachor. 31.10.91                                                         */
/*                                                                                         */
/*-----------------------------------------------------------------------------------------*/
```

```
void            dft()
{
                extern int inv;
                extern long npt;
                long     k, n;
                double WN, wk, c, s, XR[size], XI[size];
                extern complex x[size];

                WN=2*pi/npt;
                if(inv==1)
                        WN=-WN;
                for(k=0; k<npt; ++k){
                        XR[k]=0.0; XI[k]=0.0;
                        wk=k*WN;
                        for(n=0; n<npt; ++n){
                                c=cos(n*wk); s=sin(n*wk);
                                XR[k]=XR[k]+x[n+1].real*c+x[n+1].imag*s;
                                XI[k]=XI[k]-x[n+1].real*s+x[n+1].imag*c;
                        }
                        if(inv= =1){      /* divide by N for IDFT */
                                XR[k]=XR[k]/npt;
                                XI[k]=XI[k]/npt;
                        }
                }
                for (k=1; k<=npt; ++k){       /* store transformed data in x */
                      x[k].real=XR[k-1];
                      x[k].imag=XI[k-1];
                }
}
```

程序3A.3 读数据的函数，保存变换后的数据到磁盘文件上，包含常数结构定义的头文件以及包含公共声明和变量的头文件

```
/*-----------------------------------------------------------------------------------------*/
/*                                                                                         */
/*          Function to read data, in complex format, for the DFT or FFT                   */
/*                                                                                         */
/*          E C Ifeachor. Last modification: July, 1992.                                   */
/*                                                                                         */
/*-----------------------------------------------------------------------------------------*/
void            read__data()
{
                extern   long      npt;
                int        n;
                extern   complex x[size];

                for(n=0; n<size; ++n){
                        x[n].real=0;
                        x[n].imag=0;
                }
                if((in=fopen("coeff.dat", "r"))= =NULL){
                        printf("cannot open file coeff.dat\n");
                        exit(1);
                }
                fscanf(in,"%ld", &npt);
                for(n=1; n<=npt; n++){
                        fscanf(in,"%lf %lf ",&x[n].real,&x[n].imag);
                }
                fclose(in);
}

void            save__data()                                        /* file name sdata.c */
{
```

```
                long        k;
                int         k1;
                extern      long npt;
                extern      complex x[size];

                if((out=fopen("dftout.dat","w"))==NULL){
                        printf("cannot open file dftout.dat \n");
                        exit(1);
                }
                fprintf(out,"k \tXR(k) \t\tXI(k) \n");
                fprintf(out,"\n");
                for(k=1; k<=npt; ++k){
                        k1=k-1;
                        fprintf(out,"%d \t%f \t%f \n", k1, x[k].real, x[k].imag);
                }
                fclose(out);
}
/* This file contains common definitions and structures
        filename: dsp1.h
*/
#include        <stdio.h>
#include        <math.h>
#include        <dos.h>

#define size        600
#define pi          3.141592654
#define maxbits     30

typedef struct      {
        double      real;
        double      imag;
        double      modulus;
        double      angle;
        }complex;
/*
        filename: dft.h
*/
void            dft();
void            fft();
void            read__data();
void            save__data();
int             inv;
long            npt;
complex         x[size];
FILE            *in, *out, *fopen();
```

测试例 3A.1 用直接DFT程序求下列8点离散时间序列的DFT系数：

$$x(n) = \{4, 2, 1, 4, 6, 3, 5, 2\}$$

对于这一问题，由PC edlin（大多数文字处理器可以用于这一目的）创建的输入数据文件具有下列格式：

```
8
4 0
2 0
1 0
4 0
6 0
3 0
5 0
2 0
```

第一行表示数据序列的长度。

利用这一程序，数据的 DFT 如下：

k	XR(k)	XI(k)
0	27.000 000	0.000 000
1	−4.121 320	3.292 893
2	4.000 000	1.000 000
3	0.121 320	−4.707 107
4	5.000 000	−0.000 000
5	0.121 320	4.707 107
6	4.000 000	−1.000 000
7	−4.121 320	−3.292 893

测试例 3A.2　用 DFT 程序求对应于以上 DFT 系数的离散时间序列，输入数据具有下列格式：

```
8
27.000 000     0.000 000
−4.121 320     3.292 893
 4.000 000     1.000 000
 0.121 320    −4.707 107
 5.000 000    −0.000 000
 0.121 320     4.707 107
 4.000 000    −1.000 000
−4.121 320    −3.292 893
```

在响应程序的提示时要选择 IDFT 选项，程序的输出与测试例 3A.1 中的离散时间序列相同。

测试例 3A.3　第三个测试例子使用复数据序列（IEEE, 1979, Chapter 1）：

$$x(n) = Q^n,\quad n = 0, 1, \ldots, 31$$

其中 $Q = 0.9 + \mathrm{j}0.3$。

输入数据系列 $x(n)$和它的 DFT, $X(k)$分别列在表 3A.1 和表 3A.2 中。

表 3A.1　复输入数据序列

	$x(n)$	
n	实部	虚部
0	0.100000E01	0.
1	0.900000E00	0.300000E00
2	0.720000E00	0.540000E00
3	0.486000E00	0.702000E00
4	0.226800E00	0.777600E00
5	−0.291600E-01	0.767880E00
6	−0.256608E00	0.682344E00
7	−0.435650E00	0.537127E00
8	−0.553224E00	0.352719E00
9	−0.603717E00	0.151480E00
10	−0.588789E00	−0.447828E-01
11	−0.516476E00	−0.216941E00
12	−0.399746E00	−0.350190E00
13	−0.254714E00	−0.435095E00
14	−0.987144E-01	−0.467999E00
15	0.515569E-01	−0.450814E00
16	0.181645E00	−0.390265E00
17	0.280560E00	−0.296745E00
18	0.341528E00	−0.182903E00
19	0.362246E00	−0.621539E-01
20	0.344667E00	0.527352E-01
21	0.294380E00	0.150862E00

（续表）

n	x(n) 实部	x(n) 虚部
22	0.219684E00	0.224090E00
23	0.130488E00	0.267586E00
24	0.371637E-01	0.279974E00
25	−0.505440E-01	0.263125E00
26	−0.124428E00	0.221649E00
27	−0.178480E00	0.162156E00
28	−0.209279E00	0.923965E-01
29	−0.216070E00	0.203732E-01
30	−0.200575E00	−0.464851E-01
31	−0.166572E00	−0.102009E00

表 3A.2　测试例 3A.3 的变换输出

0.693972	3.499714
2.792268	8.050456
9.402964	−9.135013
1.866446	−3.833833
1.131822	−2.234158
0.904794	−1.534631
0.799557	−1.139607
0.739607	−0.882315
0.700858	−0.698566
0.673577	−0.558478
0.653112	−0.446244
0.636987	−0.352691
0.623790	−0.272085
0.612613	−0.200642
0.602885	−0.135703
0.594200	−0.075314
0.586276	−0.017948
0.578899	0.037651
0.571898	0.092607
0.565139	0.147983
0.558490	0.204882
0.551858	0.264523
0.545134	0.328363
0.538217	0.398257
0.531000	0.476679
0.523403	0.567133
0.515361	0.674850
0.506928	0.808100
0.498469	0.980906
0.491388	1.219210
0.490730	1.577083
0.517355	2.188832

3B　基-2 时域抽取 FFT 的 C 程序

这里给出的FFT程序是实现基-2时域抽取FFT（Cooley and Tukey, 1965）的C语言程序，程序计算3A.1式定义的离散时间序列的DFT或IDFT。程序由主函数dftf.c和三个函数fft()、read_data()和save_data()组成。如同直接DFT的情况，所有函数放在分开的文件里，通过主函数中的include语句在编译期间将它们组合在一起。两个函数read_data()和save_data()用来读输入数据序列和保存变换的数据，这两个函数与直接DFT用到的函数相同。主程序dftf.c和函数fft()分别列在程序3B.1和程序3B.2中。

将附录3A描述的每一个测试数据应用到FFT程序中，使用完全相同的格式产生直接DFT得到的相同结果。上述问题将留给读者，通过练习来确认这种情况。

程序 3B.1 用时域抽取 FFT 计算 DFT 的主函数

```
/*---------------------------------------------------------------------*/
/*                                                                     */
/*          Program to compute DFT coefficients using DIT FFT          */
/*          3 other functions are used                                 */
/*                                                                     */
/*          E C Ifeachor. July, 1992                                   */
/*                                                                     */
/*---------------------------------------------------------------------*/
#include    "dsp1.h"
#include    "dft.h"
main()
{
            extern long npt;
            extern int      inv;

            printf("select type of transform \n");
            printf("\n");
            printf("0     for forward DFT\n");
            printf("1     for inverse DFT\n");
            scanf("%d", &inv);
            read__data();
            fft();
            save__data();
            exit();
}
#include    "fft.c";
#include    "rdata.c";
#include    "sdata.c";
```

程序 3B.2 实现基-2 时域抽取 FFT 的 C 语言程序

```
/*---------------------------------------------------------------------*/
/*                                                                     */
/*          file name: fft.c                                           */
/*          E C Ifeachor. June, 1992                                   */
/*                                                                     */
/*          Function computes the DFT of a sequence using radix2 FFT   */
/*                                                                     */
/*                                                                     */
/*---------------------------------------------------------------------*/
void            fft()
{
                int         sign;
                long        m, irem, l, le, le1, k, ip,i,j;
                double      ur, ui, wr, wi, tr, ti, temp;
                extern      long npt;
                extern      int inv;
                extern      complex x[size];
                /* in-place bit reverse shuffling of data */

                j=1;
                for(i=1; i<npt; ++i){
                      if(i<j){
                            tr=x[j].real; ti=x[j].imag;
                            x[j].real= x[i].real;
                            x[j].imag=x[i].imag;
                            x[i].real=tr; x[i].imag=ti;
```

```
                    k=npt/2;
                    while(k<j){
                           j=j-k;
                           k=k/2;
                           }
                    }
                    else{
                           k=npt/2;
                           while(k<j){
                                  j=j-k;
                                  k=k/2;
                           }
                    }
                    j=j+k;
}
/* calculate the number of stages: m=log2(npt), and whether FFT or IFFT */
              m=0; irem=npt;
              while(irem>1){
                     irem=irem/2;
                     m=m+1;
              }
              if(inv==1)
                     sign=1;
              else
                     sign=-1;

/* perform the FFT computation for each stage */
              for(l=1; l<=m, l++){
                     le=pow(2, l);
                     le1= le/2;
                     ur=1.0; ui=0;
                     wr=cos(pi/le1);
                     wi=sign*sin(pi/le1);
                     for(j=1; j<=le1; ++j){
                            i=j;
                            while(i<=npt){
                                   ip=i+le1;
                                   tr=x[ip].real*ur-x[ip].imag*ui;
                                   ti=x[ip].imag*ur+x[ip].real*ui;
                                   x[ip].real=x[i].real-tr;
                                   x[ip].imag=x[i].imag-ti;
                                   x[i].real=x[i].real+tr;
                                   x[i].imag=x[i].imag+ti;
                                   i=i+le;
                            }
                            temp=ur*wr-ui*wi;
                            ui=ui*wr+ur*wi;
                            ur=temp;
                     }
              }
              /* If inverse fft is desired divide each coefficient by npt */
              if(inv==1){
                     for(i=1; i<=npt; ++i){
                            x[i].real=x[i].real/npt;
                            x[i].imag=x[i].imag/npt;
                     }
              }
}
```

3C MATLAB的DFT和FFT

在MATLAB和MATLAB信号处理工具箱（Signal Processing Toolbox）执行一维DFT和FFT的关键函数是dftmtx、fft和ifft，工具箱里也还包含执行离散余弦变换和二维FFT。

dftmtx函数可以用下列命令计算用矢量x表示的N点数据序列的离散傅里叶变换：

```
X = x*dftmtx(N)
```

dftmtx函数计算和返回$N \times N$复矩阵表示的旋转因子，该复矩阵与数据序列x相乘得到它的离散傅里叶变换。

DFT反变换可以使用conj命令得到：

```
x = X*conj(dftmtx(N))yN
```

fft函数用基-2 FFT算法计算一维数据序列的DFT（如果数据长度是2的幂），而ifft函数用来求DFT反变换。

MATLAB程序（参见程序3C.1）用来从文件中读出数据，然后直接计算DFT正变换或反变换，程序3C.2可以通过FFT计算DFT或DFT反变换。

程序应用的说明性例子可以在指导手册*A Practical Guide for MATLAB and C Language Implementations of DSP Algorithms*中找到。

程序3C.1　直接计算DFT的MATLAB程序

```
function DFTD

clear all;
% Program to compute DFT coefficients directly

direction=-1;  %1 - forward DFT, -1 - inverse DFT
in=fopen('datain.dat','r');
x=fscanf(in,'%g %g',[2,inf]);
fclose(in);
x=x(1,:)+x(2,:)*i; % form complex numbers

if direction==1
   y=x*dftmtx(length(x));          %compute DFT
else
   y=x*conj(dftmtx(length(x)))/length(x); %compute IDFT
end

% Save/Print the results
out=fopen('dataout.dat','w');
fprintf(out,'%g %g\n',[real(y); imag(y)]);
fclose(out);
subplot(2,1,1),plot(1:length(x),x); title('Input Signal');
subplot(2,1,2),plot(1:length(y),y); title('Output Signal');
```

程序3C.2　通过FFT计算MATLAB程序

```
function DFTF
% Program to compute DFT coefficients using DIT FFT
%

clear all;
direction=-1;  %1 - forward DFT, -1 - inverse DFT
in=fopen('dataout.dat','r');
x=fscanf(in,'%g %g',[2,inf]);
```

```
fclose(in);
x=x(1,:)+x(2,:)*i; % form complex numbers
if direction==1
  y=fft(x,length(x)) % compute FFT
else
  y=ifft(x,length(x)) % compute IFFT
end

% Save/Print the results
out=fopen('dataout.dat','w');
fprintf(out,'%g %g\n',[real(y); imag(y)]);
fclose(out);
subplot(2,1,1),plot(1:length(x),x); title('Input Signal');
subplot(2,1,2),plot(1:length(y),y); title('Output Signal');
```

附录的参考文献

Cooley J.W. and Tukey J.W. (1965) An algorithm for the machine calculation of complex Fourier series. *Mathematics Computation*, **19**(90), April, 297–301.

IEEE (1979) *Programs for Digital Signal Processing*. New York: IEEE Press.

第4章　z变换及其在信号处理中的应用

z变换对于表示、分析和设计离散时间信号与系统是十分方便且有价值的工具，它在离散时间系统中扮演了类似于拉普拉斯变换在连续时间系统中扮演的角色。

在本章，我们介绍z变换的重要内容，特别是在随后几章中要用到的内容，并且强调它在离散时间系统设计中的应用。这些应用包括用z变换描述离散时间信号和系统，以便我们能够在一定的程度上推测它们的稳定度、可视化它们的频率响应，分析数字滤波器的量化误差，以及计算离散时间系统的频率响应。大多数应用的详细内容将在随后的几章中给出。

本章以及余下的几章介绍的都是一些实际的方法，并对算法提供了必要的C语言程序和MATLAB程序，以便读者加深对内容的理解。本章的许多讨论包括线性离散时间信号与系统，所以，我们从简短地回顾这类信号与系统的特性开始讲解。

4.1　离散时间信号与系统

离散信号的值只在时间的离散点或其他一些合适的变量（如空间）有定义。正如在第1章、第2章所讨论的那样，这样的信号可以通过对连续时间信号在均匀的时间间隔的时间点nT（$n=0, 1,\ldots$）上抽样而得到，其中T是抽样周期；也可以在计算机上通过算法人工产生。离散时间信号的幅度可能具有离散值（离散时间、离散幅度），或者也有可能是连续的。

离散时间信号通常表示为数的序列：

$$x(n),\qquad n=0,1,\ldots \tag{4.1a}$$

$$x(nT),\qquad n=0,1,\ldots \tag{4.1b}$$

$$x_n,\qquad n=0,1,\ldots \tag{4.1c}$$

其中符号$x(n)$、$x(nT)$或x_n表示信号在时间n（或nT）时刻的值。为了方便起见，我们用符号$x(n)$表示序列在离散时间n和序列自身的值，除非我们希望强调二者的差别，文中的意思将是十分清楚的。由于序列并非总是时间的函数（例如它可能是空间的函数），因此在DSP的实际应用中通常省去T。有时候省去T是由于为了方便起见假定单位抽样频率（即归一化）。

离散时间系统本质上是一个数学算法，输入序列为$x(n)$，得到的输出序列为$y(n)$。离散时间系统的例子有数字控制器、数字谱分析仪和数字滤波器。离散时间系统可能是线性的或非线性的、时不变的或时变的。线性时不变系统形成了DSP中使用的一类重要的系统，数字滤波器的例子在第6章~第8章中详细讨论。

离散时间系统为线性的是指它服从叠加原理，即线性系统对两个或两个以上输入的响应等于各输入单独加到系统的响应之和。例如，如果输入$x_1(n)$加到系统上，输出为$y_1(n)$，另一个输入$x_2(n)$加到系统得到输出$y_2(n)$，那么两个信号加到系统的响应为

$$a_1x_1(n)+a_2x_2(n)\to a_1y_1(n)+a_2y_2(n) \tag{4.2}$$

其中a_1、a_2是任意常数。

离散时间系统为时不变（有时也称为位移不变）的，是指如果它的输出与输入作用的时间无关。例如，如果输入 $x(n)$给出输出 $y(n)$，那么输入 $x(n-k)$给出输出 $y(n-k)$：

$$x(n) \to y(n) \tag{4.3a}$$

$$x(n-k) \to y(n-k) \tag{4.3b}$$

即输入的时延在输出信号中只引起一个相同的时延。

LTI（线性时不变）系统输入输出之间的关系由卷积和给出，

$$y(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k) \tag{4.4}$$

其中 $h(k)$是系统的冲激响应。$h(k)$的值在时域完全定义了离散时间系统。LTI 系统是稳定的，指的是如果它的冲激响应满足条件：

$$\sum_{k=-\infty}^{\infty} |h(k)| < \infty \tag{4.5}$$

如果 $h(k)$是有限持续时间的，或者当 k 增加时 $h(k)$衰减到零，那么这个条件是满足的。稳定性的考虑将在 4.5.7 节中详细描述。

因果系统是这样一个系统，当存在输入的时候才会有输出，所有的物理系统都是因果的。一般来说，因果的离散时间序列 $x(n)$或者离散时间系统的冲激响应 $h(k)$在时间 0 之前为零，即 $x(n) = 0$ $(n < 0)$，或者 $h(k) = 0$ $(k < 0)$。本书的许多讨论都是有关实际的问题，因此都是因果系统。

4.2 z 变换

序列 $x(n)$（对所有的 n 都有效）的 z 变换定义为

$$X(z) = \sum_{n=-\infty}^{\infty} x(n)z^{-n} \tag{4.6}$$

其中 z 是复变量。

在因果系统中，$x(n)$只在 $0< n < \infty$ 非零，4.6 式化简为所谓的单边 z 变换：

$$X(z) = \sum_{n=0}^{\infty} x(n)z^{-n} \tag{4.7}$$

很显然，z变换是有无限项的幂级数，所以并非对所有的z收敛，z变换收敛的区域称为收敛域（ROC），在这个区域里 $X(z)$的值是有限的。收敛域由 $x(n)$的性质确定，或者等价地由 $X(z)$的性质确定，下面的一组例子说明了这一点。

例 4.1 对于图 4.1 给出的离散时间序列，求 z 变换和收敛域。

(1) 图 4.1(a)的序列是非因果的，因为 $n < 0$ 时 $x(n)$非零，但它是有限持续时间的，序列的值为：$x(-6) = 0, x(-5) = 1, x(-4) = 3, x(-3) = 5, x(-2) = 3, x(-1) = 1, x(0) = 0$。由 4.6 式，序列的 z 变换为

$$\begin{aligned} X_1(z) &= \sum_{n=-\infty}^{\infty} x(n)z^{-n} \\ &= z^5 + 3z^4 + 5z^3 + 3z^2 + z \end{aligned}$$

很容易验证，当 $z=-\infty$ 时，$X(z)$变成了无穷大，因此，ROC 包含了 z 平面上除了在 $z=-\infty$ 外的任何地方。

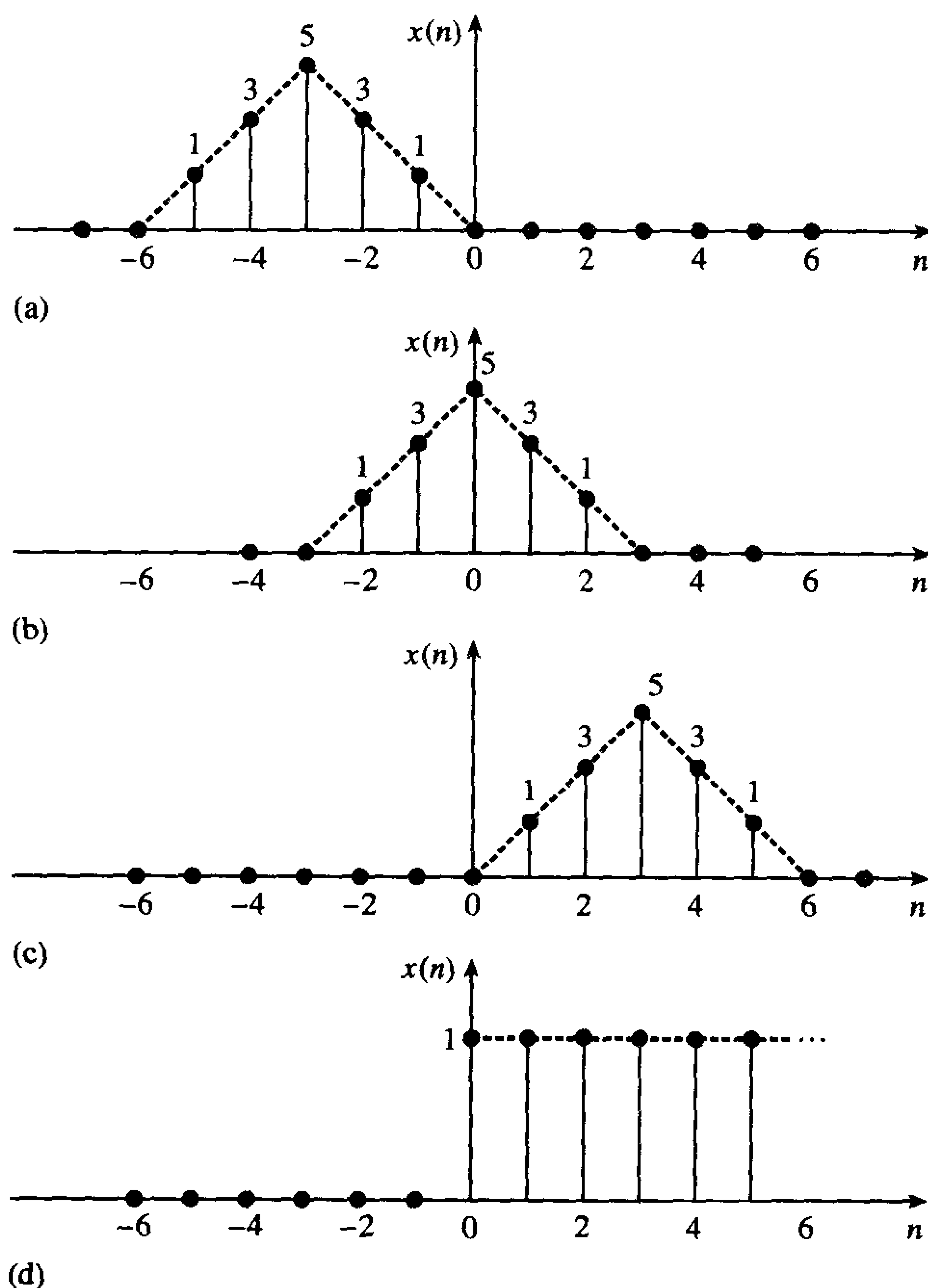


图 4.1　因果和非因果的离散时间序列

(2) 图 4.1(b)的序列是非因果的，它具有有限持续时间，并且是双边的。序列的值是 $x(3)=0$, $x(-2)=1, x(-1)=3, x(0)=5, x(1)=3, x(2)=1, x(3)=0$。由 4.6 式，$z$ 变换为

$$X_2(z) = \sum_{n=-\infty}^{\infty} x(n)z^{-n}$$

$$= z^2 + 3z + 5 + 3z^{-1} + z^{-2}$$

很明显，如果 $z=0$ 或者 $z=\infty$，那么 $X(z)$的值是无穷大。因此收敛域是除 $z=0$ 和 $z=\infty$ 的所有地方。

(3) 图 4.1(c)表示因果的有限持续时间序列，其值为 $x(0)=0, x(1)=1, x(2)=3, x(3)=5, x(4)=3,$ $x(5)=1, x(6)=0$，z 变换为

$$X_3(z) = \sum_{n=-\infty}^{\infty} x(n)z^{-n}$$

$$= z^{-1} + 3z^{-2} + 5z^{-3} + 3z^{-4} + z^{-5}$$

在这种情况下，当 $z=0$ 时 $X(z)=\infty$，因此，收敛域是除 $z=0$ 外的所有地方。

(4) 图 4.1(d)的离散时间序列定义为

$$x(n) = 1 \quad 0 \leqslant n \leqslant \infty$$
$$= 0 \quad n < 0$$

很显然，它是无限持续时间的因果序列。由4.6式，序列的z变换为

$$X(z) = \sum_{n=-\infty}^{\infty} x(n)z^{-n}$$
$$= \sum_{n=0}^{\infty} z^{-n}$$
$$= 1 + z^{-1} + z^{-2} + \ldots$$

这是一个公比为z^{-1}的几何级数，如果$|z^{-1}| < 1$，或者等价于如果$|z| > 1$，这个级数是收敛的。这样，倘若$|z| > 1$，我们可以用闭合形式表示$X(z)$为

$$X(z) = 1 + z^{-1} + z^{-2} + \ldots$$
$$= 1/(1 - z^{-1}) = z/(z - 1) \tag{4.8}$$

在这种情况下，z变换在中心为原点的单位圆外处处有效，圆的外部是收敛域（参见图4.2）。我们很容易验证，当$|z| > 1$时，$X(z)$收敛，而当$|z| < 1$时，$X(z)$发散。例如，如果我们令$z = 2$（单位圆外），那么，4.8式的级数和为

$$X(z) = 1 + 1/2 + (1/2)^2 + (1/2)^3 + \ldots = 2/(2 - 1) = 2$$

很显然，它是公比为1/2的几何级数，首项为1，无穷和为2/(2–1) = 2。另一方面，如果$z = 1/2$（单位圆内），4.8式的级数变成

$$X(z) = 1 + 1/0.5 + (1/0.5)^2 + (1/0.5)^3 + \ldots = 1 + 2 + 4 + 8 + \ldots$$

显然它是发散的。在图4.2中，收敛域（阴影部分）由圆$|z| = 1$作为边界，$X(z)$的极点的半径也为1。使$X(z) = \infty$的z称为$X(z)$的极点，而使$X(z) = 0$的z称为$X(z)$的零点。

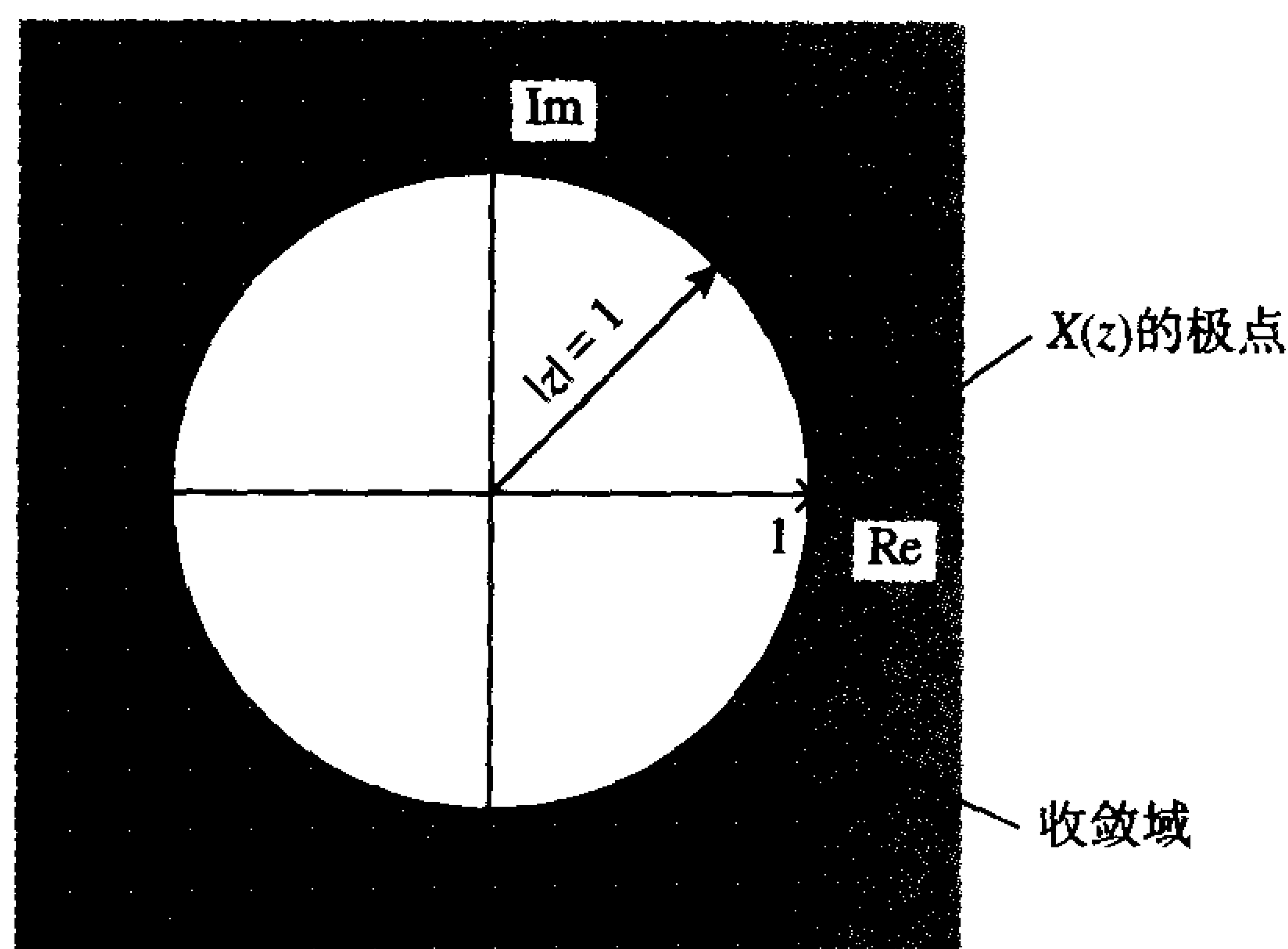


图4.2 例4.1部分(4)的收敛域

从上面的例子我们可以推断，有限持续时间因果序列的z变换在某个圆外处处收敛（除了$z = 0$），圆的半径是具有最大半径的极点的半径。对于稳定的系统，ROC总是应该包含单位圆，对于具有频率响应的系统是很重要的。

常用序列的z变换用闭合形式表示是有用的。表4.1给出了常用序列的z变换，这个表在求z反变换时是很有用的。

表 4.1　常用序列的 z 变换

序号	离散时间序列 $x(n)$（$n \geqslant 0$）	z 变换 $X(z)$	$X(z)$的收敛域
1	$k\delta(n)$	k	任何位置
2	k	$\frac{kz}{z-1}$	$\lvert z\rvert > 1$
3	kn	$\frac{kz}{(z-1)^2}$	$\lvert z\rvert > 1$
4	kn^2	$\frac{kz(z+1)}{(z-1)^3}$	$\lvert z\rvert > 1$
5	$ke^{-\alpha n}$	$\frac{kz}{z-e^{-\alpha}}$	$\lvert z\rvert > e^{-\alpha}$
6	$kne^{-\alpha n}$	$\frac{kze^{-\alpha}}{(z-e^{-\alpha})^2}$	$\lvert z\rvert > e^{-\alpha}$
7	$1-e^{-\alpha n}$	$\frac{z(1-e^{-\alpha})}{z^2-z(1+e^{-\alpha})+e^{-\alpha}}$	$\lvert z\rvert > e^{-\alpha}$
8	$\cos(\alpha n)$	$\frac{z(z-\cos\alpha)}{z^2-2z\cos\alpha+1}$	$\lvert z\rvert > 1$
9	$\sin(\alpha n)$	$\frac{z\sin\alpha}{z^2-2z\cos\alpha+1}$	$\lvert z\rvert > 1$
10	$e^{-\alpha n}\sin(\alpha n)$	$\frac{ze^{-\alpha}\sin\alpha}{z^2-2e^{-\alpha}z\cos\alpha+e^{-2\alpha}}$	$\lvert z\rvert > e^{-\alpha}$
11	$e^{-\alpha n}\cos(\alpha n)$	$\frac{ze^{-\alpha}(ze^{\alpha}-\cos\alpha)}{z^2-2ze^{-\alpha}\cos\alpha+e^{-2\alpha}}$	$\lvert z\rvert > e^{-\alpha}$
12	$\cosh(\alpha n)$	$\frac{z^2-z\cosh\alpha}{z^2-2z\cosh\alpha+1}$	$\lvert z\rvert > \cosh\alpha$
13	$\sinh(\alpha n)$	$\frac{z\sinh\alpha}{z^2-2z\cosh\alpha+1}$	$\lvert z\rvert > \sinh\alpha$
14	$k\alpha^n$	$\frac{kz}{z-\alpha}$	$\lvert z\rvert > \alpha$
15	$kn\alpha^n$	$\frac{k\alpha z}{(z-\alpha)^2}$	$\lvert z\rvert > \alpha$
16	$2\lvert c\rvert\lvert p\rvert^n\cos(n\angle p+\angle c)$	$\frac{cz}{z-p}+\frac{c^*z}{z-p^*}$	

k 和 α 是常数，c 是复数。

4.3　z 反变换

z 反变换（IZT）允许我们在给定离散时间序列 $x(n)$的 z 变换时恢复 $x(n)$，IZT 在 DSP 中是特别有用的，例如求数字滤波器的冲激响应。z 反变换定义为

$$x(n) = Z^{-1}[X(z)] \tag{4.9}$$

其中 $X(z)$是 $x(n)$的 z 变换，Z^{-1} 是 z 反变换的符号。

假定在 4.7 式中的因果序列的 z 变换 $X(z)$可以用幂级数展开为

$$X(z)=\sum_{n=0}^{\infty}x(n)z^{-n}$$
$$=x(0)+x(1)z^{-1}+x(2)z^{-2}+x(3)z^{-3}+\dots \tag{4.10}$$

可以看出，$x(n)$的值是 z^{-n}（$n=0, 1,\dots$）的系数，因此，通过观察可以直接求得 $x(n)$。实际上，$X(z)$常常可以表示成两个用 z^{-1} 或者等价地用 z 表示的多项式之比：

$$X(z)=\frac{b_0+b_1z^{-1}+b_2z^{-2}+\dots+b_Nz^{-N}}{a_0+a_1z^{-1}+a_2z^{-2}+\dots+a_Mz^{-M}} \tag{4.11}$$

在这种形式中，z 反变换 $x(n)$可以用几种方法得到：

(1) 幂级数展开法

(2) 部分分式展开法

(3) 留数法

每种方法都有自身的优点和缺点。严格地从数学上来讲，留数法或许是最好的方法，然而，幂级数法是最适合于计算机实现的方法。

在下面几节，我们将依次描述以上三种方法，利用一些数值计算的例子来说明所含的原理。在附录中，我们描述了方法(1)和方法(2)计算z反变换的C语言程序清单，并且给出了几个说明性的数值计算例子。

4.3.1 幂级数法

给定一个例3.11中的因果序列的z变换$X(z)$，通过长除法（有时也叫综合除法），它可以用一个 z^{-1} 或 z 的无穷级数展开：

$$X(z)=\frac{b_0+b_1z^{-1}+b_2z^{-2}+\dots+b_Nz^{-N}}{a_0+a_1z^{-1}+a_2z^{-2}+\dots+a_Mz^{-M}}$$
$$=x(0)+x(1)z^{-1}+x(2)z^{-2}+x(3)z^{-3}+\dots \tag{4.12}$$

在这种方法中，$X(z)$的分子和分母首先用 z 的幂递减或 z^{-1} 的幂递增的形式表示，由长除法得到商。下面举例说明此方法。

例 4.2 给定下列因果 LSI 系统的 z 变换，

$$X(z)=\frac{1+2z^{-1}+z^{-2}}{1-z^{-1}+0.3561z^{-2}}$$

通过长除法将其展开成幂级数形式来求 z 反变换。

解：

首先，将 $X(z)$的分子分母用 z^{-1} 的幂递增的形式表示，然后执行通常的长除法，

$$\begin{array}{r|l}
 & 1+3z^{-1}+3.6439z^{-2}+2.5756z^{-3}+\dots \\
\hline
1-z^{-1}+0.3561z^{-2} & 1+2z^{-1}+z^{-2} \\
 & \underline{1-z^{-1}+0.3561z^{-2}} \\
 & 3z^{-1}+0.6439z^{-2} \\
 & \underline{3z^{-1}-3z^{-2}+1.0683z^{-3}} \\
 & 3.6439z^{-2}-1.0683z^{-3} \\
 & \underline{3.6439z^{-2}-3.6439z^{-3}+1.2975927z^{-4}} \\
 & 2.5756z^{-3}-1.2975927z^{-4}
\end{array}$$

此外，我们也可以将$X(z)$的分子分母用z的幂递减形式表示，然后执行长除法，

$$\frac{z^2+2z+1}{z^2-z+0.3561}$$

$$1+3z^{-1}+3.6439z^{-2}+2.5756z^{-3}+\ldots$$

$$z^2-z+0.3561 \,\big)\, z^2+2z+1$$

$$\underline{z^2-\ z+0.3561}$$

$$3z+0.6439$$

$$\underline{3z-3+1.0683z^{-1}}$$

$$3.6439-3.643\,91z^{-1}+1.297\,592\,7z^{-2}$$

$$2.5756z^{-1}-1.297\,592\,7z^{-2}$$

两种方法都把z变换展开成了熟悉的幂级数形式，即

$$X(z)=1+3z^{-1}+3.6439z^{-2}+2.5756z^{-3}+\ldots$$

那么，z反变换就可以直接写出：

$$x(0)=1;\ x(1)=3;\ x(2)=3.6439;\ x(3)=2.5756;\ \ldots$$

长除法可以重新描述如下（参见附录4A），$x(n)$的值递归地得到

$$\begin{aligned}
x(0)&=b_0/a_0\\
x(1)&=[b_1-x(0)a_1]/a_0\\
x(2)&=[b_2-x(1)a_1-x(0)a_2]/a_0\\
&\ \vdots\\
x(n)&=\left[b_n-\sum_{i=1}^{n}x(n-i)a_i\right]\Big/a_0,\quad n=1,2,\ldots
\end{aligned} \tag{4.13a}$$

其中

$$x(0)=b_0/a_0 \tag{4.13b}$$

我们重复前面的例子来说明递归方法。

例4.3　用递归法求z反变换$x(n)$的前四项，假定z变换$X(z)$与例4.2相同，即

$$X(z)=\frac{1+2z^{-1}+z^{-2}}{1-z^{-1}+0.3561z^{-2}}$$

解：

比较$X(z)$的系数和4.12式中一般变换的相关系数，可以有

$$a_0=1,\ a_1=2,\ a_2=1,\ b_0=1,\ b_1=-1,\ b_2=0.3561;\ N=M=2$$

由4.13式，我们有

$$\begin{aligned}
x(0)&=b_0/a_0=1\\
x(1)&=[b_1-x(0)a_1]/a_0=[2-1\times(-1)]=3\\
x(2)&=[b_2-x(1)a_1-x(0)a_2]=1-3\times(-1)-1\times0.3561=3.6439\\
x(3)&=[b_3-x(2)a_1-x(1)a_2+x(0)a_3]\\
&=0-x(2)a_1-x(1)a_2=0-3.6439\times(-1)-3\times0.3561=2.5756
\end{aligned}$$

即 z 反变换的前四项为

$$x(0)=1,\ x(1)=3,\ x(2)=3.6439,\ x(3)=2.5756$$

可以看出，递归的和直接的长除法都得出相同的结果。

4.13 的递推式通过下面的 C 语言代码可以很容易在计算机上实现：

```
x[0]=B[0]/A[0];
for(n=1;n<=npt;++n){
        sum=0;
        k=n;
        if(n>M)
                k=M;
        for(i=1;i<=k; ++i){
                sum=sum+x[n-i]*A[i];
        }
        x[n]=(B[n]-sum)/A[0];
}
```

在程序代码中，M 是分母多项式的阶数，npt 是 IZT 的数据点数，分子和分母多项式假定是按 z^{-1} 的升幂表示的。在附录 4B 和附录 4D 分别给出了根据以上代码计算 IZT 的 MATLAB 程序和 C 语言程序。

4.3.2 部分分式展开方法

在这种方法中，z 变换首先展开成简单的部分分式之和。每一个部分分式的 z 反变换从像表 4.1 那样的变换表中得到，然后对给出的整个 z 反变换求和。在许多实际情况下，z 变换是多项式之比，多项式用 z 或 z^{-1} 表示，一般具有如下熟悉的形式：

$$X(z)=\frac{b_0+b_1z^{-1}+b_2z^{-2}+\ldots+b_Nz^{-N}}{a_0+a_1z^{-1}+a_2z^{-2}+\ldots+a_Mz^{-M}} \tag{4.14}$$

如果 $X(z)$ 的极点是一阶的，且 $N=M$，那么 $X(z)$ 可以展开成

$$\begin{aligned}X(z)&=B_0+\frac{C_1}{1-p_1z^{-1}}+\frac{C_2}{1-p_2z^{-1}}+\ldots+\frac{C_M}{1-p_Mz^{-1}}\\&=B_0+\frac{C_1z}{z-p_1}+\frac{C_2z}{z-p_2}+\ldots+\frac{C_Mz}{z-p_M}=B_0+\sum_{k=1}^{M}\frac{C_kz}{z-p_k}\end{aligned} \tag{4.15}$$

其中 p_k 是 $X(z)$ 的极点（假定不同），C_k 是部分分式系数，且

$$B_0=b_N/a_N \tag{4.16}$$

C_k 也称为 $X(z)$ 的留数，参见 4.3.3 节。

如果 4.14 式中分子的阶数小于分母的阶数，即 $N<M$，那么 B_0 将为零。如果 $N>M$，那么必须首先用长除法将 $X(z)$ 化简，使 $N\leqslant M$，分子分母用 z^{-1} 的升幂表示，余数表示成 4.15 式的形式。

在等式 4.15 两边乘以 $(z-p_k)/z$，然后令 $z=p_k$，就可以得到与极点 p_k 有关的系数 C_k：

$$C_k=\frac{X(z)}{z}(z-p_k)\bigg|_{z=p_k} \tag{4.17}$$

如果 $X(z)$ 包含一个或者多个多阶极点（即极点是重合的），考虑这种情况要求增加额外的项。例如，如果 $X(z)$ 在 $z=p_k$ 处包含一个 m 阶极点，那么部分分式展开必须包括如下形式的项：

$$\sum_{i=1}^{m} \frac{D_i}{(z-p_k)^i} \tag{4.18a}$$

系数 D_i 可从如下关系得到：

$$D_i = \frac{1}{(m-i)!} \frac{\mathrm{d}^{m-i}}{\mathrm{d}z^{m-i}} \left[(z-p_k)^m \frac{X(z)}{z} \right]_{z=p_k} \tag{4.18b}$$

由部分分式展开法计算 z 反变换最好是用例子加以说明。

例 4.4　$X(z)$ 包含简单的一阶极点　求下列 z 变换的反变换：

$$X(z) = \frac{z^{-1}}{1-0.25z^{-1}-0.375z^{-2}}$$

解：

为了简单起见，我们首先通过给分子分母乘以 z^2（z 的最高幂），使 z 变换用 z 的正幂表示：

$$X(z) = \frac{z}{z^2-0.25z-0.375} = \frac{z}{(z-0.75)(z+0.5)}$$

$X(z)$ 在 $z=0.75$ 和 $z=-0.5$ 包含一阶极点（即在每个极点位置只有一个极点）。由于分子的阶数小于分母的阶数（$N<M$），部分分式展开具有如下形式：

$$X(z) = \frac{z}{(z-0.75)(z+0.5)} = \frac{C_1 z}{z-0.75} + \frac{C_2 z}{z+0.5} \tag{4.19}$$

为了能够容易求出 C_k 的值，我们在两边同时除以 z，

$$\frac{X(z)}{z} = \frac{z}{z(z-0.75)(z+0.5)} = \frac{C_1}{z-0.75} + \frac{C_2}{z+0.5} \tag{4.20}$$

为了得到 C_1，我们简单地在 4.20 式两边乘以 $z-0.75$，且令 $z=0.75$，得

$$\frac{(z-0.75)X(z)}{z} = \frac{\cancel{(z-0.75)}}{\cancel{(z-0.75)}(z+0.5)} = C_1 + \frac{C_2(z-0.75)}{z+0.5}$$

$$C_1 = \left.\frac{1}{z+0.5}\right|_{z=0.75} = \frac{1}{0.75+0.5} = \frac{4}{5}$$

类似地，C_2 为

$$C_2 = \left.\frac{(z+0.5)X(z)}{z}\right|_{z=-0.5}$$

$$= \left.\frac{\cancel{(z+0.5)}}{(z-0.75)\cancel{(z+0.5)}}\right|_{z=-0.5} = \frac{1}{-0.5-0.75} = -\frac{4}{5}$$

在 4.19 式中应用 C_1 和 C_2，我们有

$$X(z) = \frac{(4/5)z}{z-0.75} - \frac{(4/5)z}{z+0.5} \tag{4.21}$$

由 z 变换表（表 4.1 中的第 14 项），4.21 式右边每一项的 z 反变换为

$$Z^{-1}\left[\frac{(4/5)z}{z-0.75}\right] = \frac{4(0.75)^n}{5}$$

$$Z^{-1}\left[\frac{-(4/5)z}{z+0.5}\right] = \frac{-4(-0.5)^n}{5}$$

希望的z反变换$x(n)$是两个z反变换之和：

$$x(n)=\frac{4}{5}[(0.75)^n-(-0.5)^n],\quad n>0$$

例 4.5　$X(z)$包含一阶复共轭极点　用部分分式展开法求用下列z变换表示的离散信号$x(n)$：

$$X(z)=\frac{1+2z^{-1}+z^{-2}}{1-z^{-1}+0.3561z^{-2}}$$

解：

首先，$X(z)$用z的正幂表示：

$$X(z)=\frac{N(z)}{D(z)}=\frac{z^2+2z+1}{z^2-z+0.3561}$$

利用下面的公式，$X(z)$的极点通过求解二次方程式$D(z)=z^2-z+0.3561=0$得到，

$$p_1=\frac{-b+(b^2-4ac)^{1/2}}{2a}$$
$$p_2=\frac{-b-(b^2-4ac)^{1/2}}{2a}\tag{4.22}$$

其中a和b分别是z^2和z的系数，c是常数项。当$a=1$、$b=-1$和$c=0.3561$时，极点为

$$p_1=\frac{-1+(1-4\times0.3561)^{1/2}}{2}$$
$$=0.5+0.3257\text{j}=re^{\text{j}\theta}$$
$$p_2=p_1^*=0.5-0.3257\text{j}=re^{-\text{j}\theta}$$

其中$r=0.5967$，$\theta=33.08^\circ$。因此，我们可以根据它的极点将$X(z)$表示为

$$X(z)=\frac{z^2+2z+1}{(z-p_1)(z-p_1^*)}$$

由于$X(z)$的分子和分母具有相同的阶，部分分式展开的形式为

$$\frac{X(z)}{z}=\frac{B_0}{z}+\frac{C_1}{z-p_1}+\frac{C_2}{z-p_1^*}\tag{4.23}$$

由4.16式，$B_0=1/0.3561=2.8082$。为了求C_1，我们在4.23式的两边乘以$z-p_1$，然后令$z=p_1$：

$$\frac{(z-p_1)X(z)}{z}=\frac{B_0(z-p_1)}{z}+C_1+\frac{C_2(z-p_1)}{z-p_2}\bigg|_{z=p_1}$$

于是

$$C_1=\frac{(z-p_1)X(z)}{z}=\frac{\cancel{(z-p_1)}(z^2+2z+1)}{z\cancel{(z-p_1)}(z-p_2)}\bigg|_{z=p_1=re^{\text{j}\theta}}$$
$$=\frac{(re^{\text{j}\theta})^2+2re^{\text{j}\theta}+1}{re^{\text{j}\theta}(re^{\text{j}\theta}-re^{-\text{j}\theta})}\tag{4.24}$$

其中$r=0.5967$，$\theta=33.08^\circ$。经处理和简化后，我们有

$$C_1=\frac{2.1439+0.97719\text{j}}{-0.2122+0.3257\text{j}}$$
$$=-0.9040999-5.992847\text{j}$$
$$=6.06066\angle-98.58^\circ$$

由于p_1和p_2是复共轭对，那么

$$C_2 = C_1^* = -0.904\,099\,9 + 5.992\,847\text{j} = 6.060\,66\angle 98.58°$$

因此，z 变换可以表示为（由 4.23 式）

$$X(z) = 2.8082 + \frac{C_1 z}{z - p_1} + \frac{C_2 z}{z - p_1^*} \tag{4.25}$$

其中

$$p_1 = 0.5 + 0.3257\text{j} \qquad p_2 = 0.5 - 0.3257\text{j}$$

$$C_1 = -0.9041 - 5.599\,28\text{j} \qquad C_2 = -0.9041 + 5.599\,28\text{j}$$

根据z变换表 4.1 的第 1 项和第 16 项，4.25 式右边项的z反变换为

$$Z^{-1}(2.8082) = 2.8082u(n)$$

$$Z^{-1}\left[\frac{C_1 z}{z - p_1} + \frac{C_2 z}{z - p_1^*}\right] = 2 \times 6.060\,66(0.5967)^n \cos(33.08n - 98.58°)$$

$$= 12.1213(0.5967)^n \cos(33.08n - 98.58°)$$

因此，离散信号变成

$$x(n) = 2.8082u(n) + 12.1213(0.5967)^n \cos(33.08n - 98.58°), \quad n \geqslant 0$$

计算$x(n)$在$n = 0, 1, 2$的值，然后将结果与幂级数展开法得到的值进行比较，这是部分分式展开的有用的检查方法。由$x(n)$的表达式，我们求得

$$x(0) = 2.8082 - 1.808\,38 = 1;\ x(1) = 2.999\,59 = 3;\ x(2) = 3.6436$$

将上面的值与例 4.3 中用幂级数法得到的结果进行比较。

例 4.6　$X(z)$包含一个二阶极点　求具有下列z变换的离散时间序列$x(n)$：

$$X(z) = \frac{z^2}{(z - 0.5)(z - 1)^2}$$

解：

$X(z)$在$z = 0.5$有一个一阶极点，在$z = 1$有一个二阶极点。在这种情况下，部分分式展开具有下列形式：

$$X(z) = \frac{C}{z - 0.5} + \frac{D_1}{z - 1} + \frac{D_2}{(z - 1)^2} \tag{4.26}$$

为了求C，我们像前一个例子一样在 4.26 式两边乘以z–0.5，然后再令$z = 0.5$，并且计算表达式：

$$C = \left.\frac{\cancel{(z - 0.5)}z^2}{z\cancel{(z - 0.5)}(z - 1)^2}\right|_{z=0.5}$$

$$= 0.5/(0.5 - 1)^2 = 2$$

为了求D_1，我们使用 4.18b 式，令$i = 1$、$m = 2$，于是有

$$D_1 = \frac{\text{d}}{\text{d}z}\left[\frac{(z-1)^2 X(z)}{z}\right]_{z=1} = \frac{\text{d}}{\text{d}z}\left[\frac{\cancel{(z-1)^2} z^{\cancel{2}}}{\cancel{z}(z - 0.5)\cancel{(z-1)^2}}\right]_{z=1}$$

$$= \frac{\text{d}}{\text{d}z}\left(\frac{z}{z - 0.5}\right)_{z=1} = \left.\frac{z - 0.5 - z}{(z - 0.5)^2}\right|_{z=1} = -2$$

类似地，D_2 由 4.18b 式令 $i=2$、$m=2$ 得到：

$$D_2 = \left.\frac{(z-1)^2 X(z)}{z}\right|_{z=1} = \left.\frac{(z-1)^2 z^2}{z(z-0.5)(z-1)^2}\right|_{z=1}$$
$$= 1/(1-0.5) = 2$$

组合以上结果，$X(z)$ 变成

$$X(z) = \frac{2z}{z-0.5} - \frac{2z}{z-1} + \frac{2z}{(z-1)^2}$$

右边每一项的 z 反变换由表 4.1 得到，然后再求和给出 $x(n)$：

$$x(n) = 2(0.5)^n - 2 + 2n = 2[(n-1)+(0.5)^n], \quad n \geqslant 0 \tag{4.27}$$

读者可以通过比较由幂级数方法得到的前几个值来验证结果的正确性。

我们承认部分分式展开法是非常繁琐的，除了一些简单的情况，一般情况下有可能出错。在附录中描述了对于一阶极点使用部分分式展开法计算 z 反变换的 C 语言和 MATLAB 程序。

4.3.3 留数法

在这种方法中，IZT 是通过计算下面的围线积分得到的，

$$x(n) = \frac{1}{2\pi \mathrm{j}} \oint_C z^{n-1} X(z)\,\mathrm{d}z \tag{4.28}$$

其中 C 是围绕 $X(z)$ 的所有极点的积分路径。对于有理多项式，4.28 式的围线积分是采用在复变函数中称为柯西（Cauchy）留数定理（Mathews, 1982）的基本结果来计算的：

$$x(n) = \frac{1}{2\pi \mathrm{j}} \oint_C z^{n-1} X(z)\,\mathrm{d}z \tag{4.29}$$
$$= z^{n-1}X(z)\text{在 } C \text{ 内的所有极点的留数之和}$$

在上一节，我们阐述了部分分式展开的系数，C_k 也称为 $X(z)$ 的留数，并给出了求其值的一种方法。要记住的关键点是，每个留数 C_k 与极点 p_k 有关。在现有的方法中，在极点处 $z^{n-1}X(z)$ 的留数（不是 $X(z)$ 的留数）为

$$\mathrm{Res}[F(z), p_k] = \frac{1}{(m-1)!}\frac{\mathrm{d}^{m-1}}{\mathrm{d}z^{m-1}}[(z-p_k)F(z)]_{z=p_k} \tag{4.30}$$

其中 $F(z) = z^{n-1}X(z)$，m 是在 p_k 极点的阶数，$\mathrm{Res}[F(z), p_k]$ 是 $F(z)$ 在 $z=p_k$ 处的留数。对于简单（不同的）极点，4.30 式化简为

$$\mathrm{Res}[F(z), p_k] = (z-p_k)F(z) = (z-p_k)z^{n-1}X(z)\big|_{z=p_k} \tag{4.31}$$

例 4.7 用留数法求对应于下面的 z 变换的离散信号：

$$X(z) = \frac{z}{(z-0.75)(z+0.5)}$$

假定 C 是圆 $|z|=1$。

解：

这个问题与例 4.4 相同，$X(z)$ 用因式分解的形式给出，

$$X(z) = \frac{z}{(z-0.75)(z+0.5)}$$

如果我们令 $F(z) = z^{n-1}X(z)$，那么

$$F(z) = \frac{z^{n-1}z}{(z-0.75)(z+0.5)} = \frac{z^n}{(z-0.75)(z+0.5)}$$

$F(z)$在 $z = 0.75$ 和 $z = -0.5$ 有极点。图 4.3 给出了围线图，极点的位置在图中用“×”表示。两个极点都在围线内（单位圆）。由 4.29 式，由于极点是一阶极点，z 反变换为

$$x(n) = \text{Res}\,[F(z), 0.75] + \text{Res}\,[F(z), -0.5]$$

由于极点是一阶的，我们应该采用 4.31 式。于是

$$\begin{aligned}\text{Res}\,[F(z), 0.75] &= (z-0.75)F(z)\big|_{z=0.75} \\ &= \frac{(z-0.75)z^n}{(z-0.75)(z+0.5)}\bigg|_{z=0.75} \\ &= \frac{(0.75)^n}{0.75+0.5} \\ &= \frac{4}{5}(0.75)^n\end{aligned}$$

$$\begin{aligned}\text{Res}\,[F(z), -0.5] &= (z+0.5)F(z)\big|_{z=-0.5} \\ &= \frac{(z+0.5)z^n}{(z-0.75)(z+0.5)}\bigg|_{z=-0.5} \\ &= -\frac{4}{5}(-0.5)^n\end{aligned}$$

z 反变换是在 $z = 0.75$ 和 $z = -0.5$ 处的留数和：

$$x(n) = (4/5)[(0.75)^n - (-0.5)^n]$$

上式与部分分式展开法得到的结果相同。

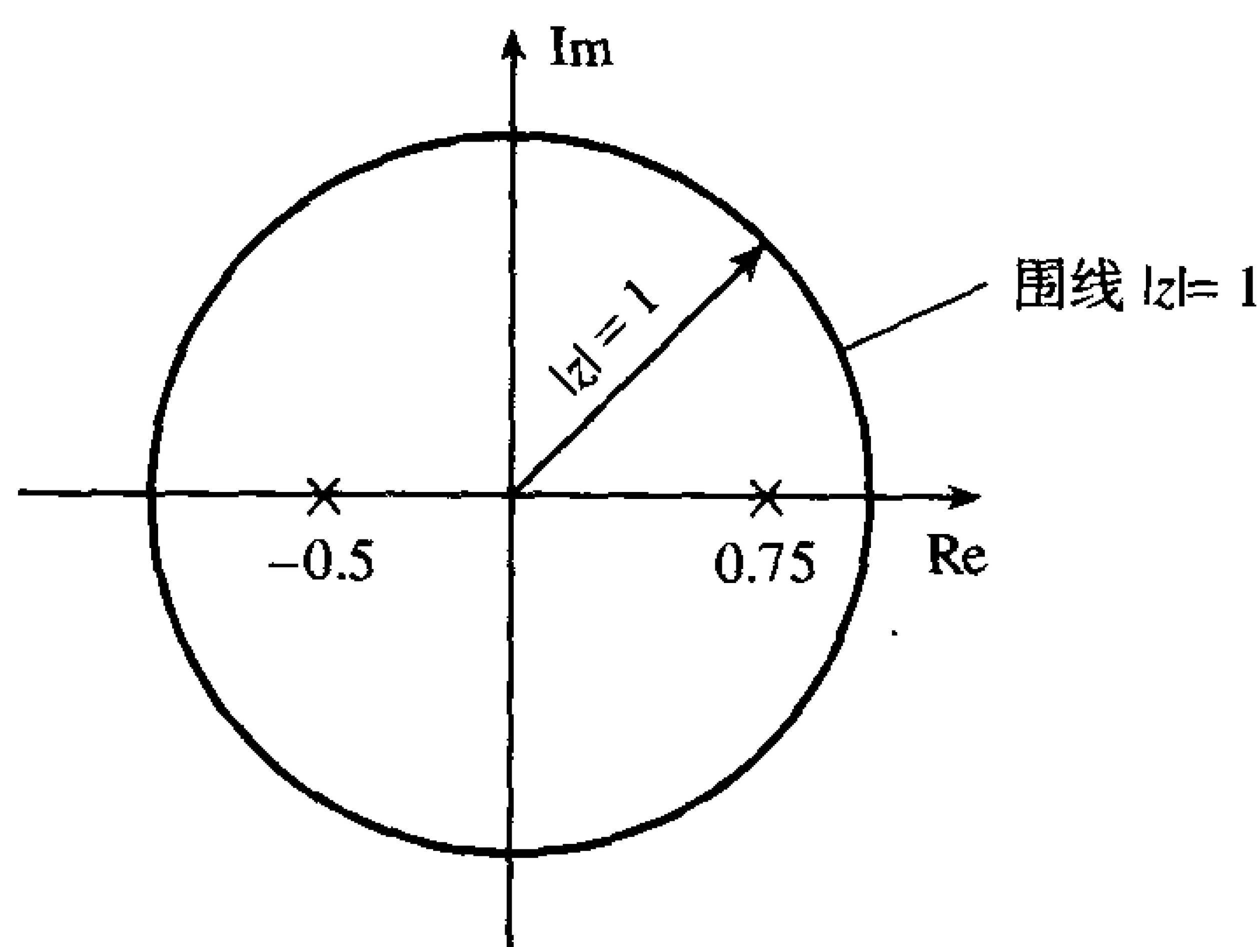


图 4.3　标明了 $X(z)$的极点的积分围线图

例 4.8　$X(z)$的极点是复共轭极点　给定下列 z 变换，用留数法求 z 反变换：

$$X(z)=\frac{z^2+2z+1}{z^2-z+0.3561}$$

解：

$X(z)$因式分解为

$$X(z)=\frac{z^2+2z+1}{(z-p_1)(z-p_2)}$$

其中$p_1=0.5+0.3557\mathrm{j}$，$p_2=0.5-0.3557\mathrm{j}$，即$p_2=p_1^*$，为了求z的反变换，我们计算如下$F(z)$的留数：

$$F(z)=z^{n-1}X(z)=\frac{z^{n-1}(z^2+2z+1)}{z^2-z+0.3561}=\frac{z^n(z^2+2z+1)}{z(z^2-z+0.3561)}$$

$F(z)$与$X(z)$有相同的极点，即$z=p_1$和$z=p_2$，当$n=0$时加上一个极点$z=0$。图4.4画出了极点位置的积分围线图，所有极点都在围线内。在$z=0$的极点当$n>0$时不存在，所以，我们需要分别考虑两种情况。

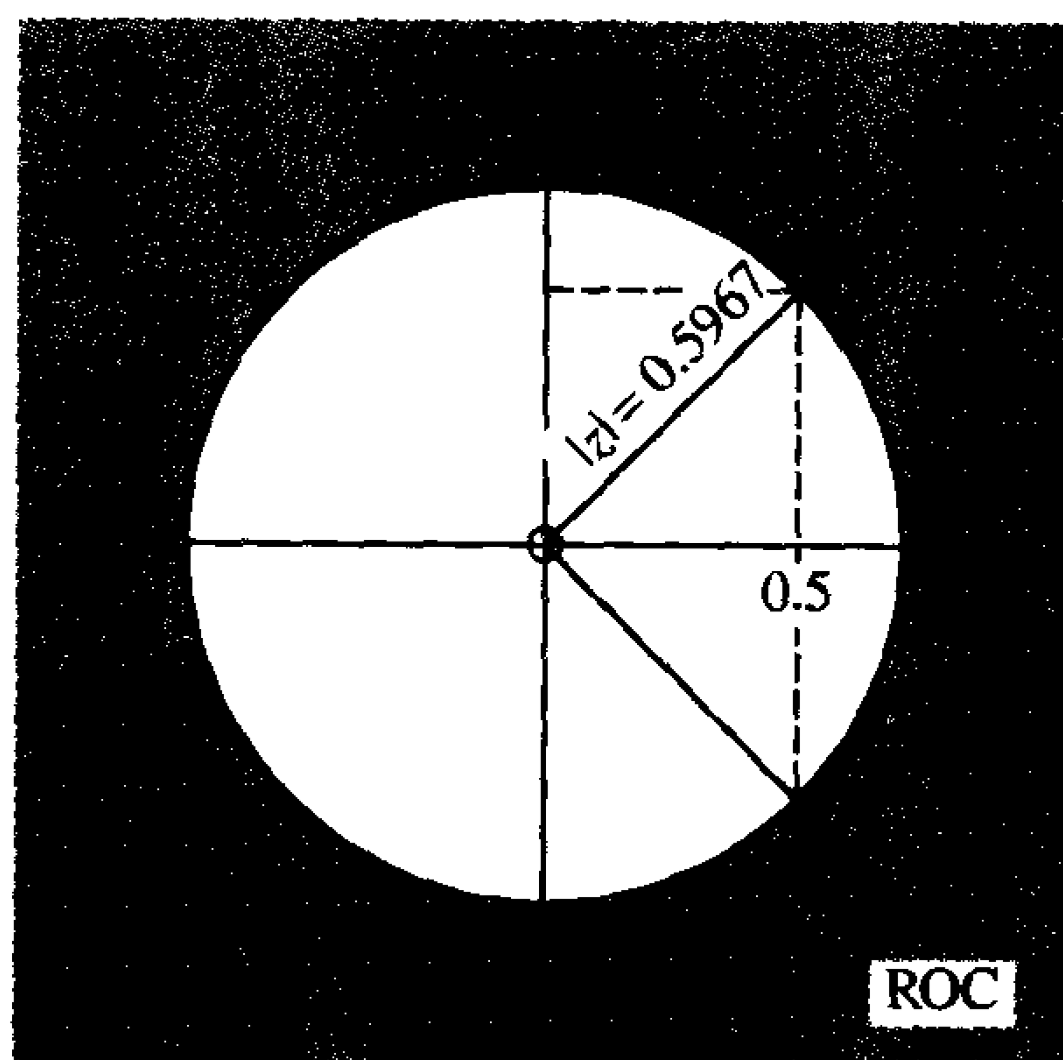


图4.4　例4.8标明的ROC的积分围线

当$n=0$时，$F(z)$化简为

$$F(z)=\frac{z^2+2z+1}{z(z^2-z+0.3561)}$$

和

$$x(0)=\mathrm{Res}[F(z),0]+\mathrm{Res}[F(z),p_1]+\mathrm{Res}[F(z),p_2]$$

因此

$$\mathrm{Res}[F(z),0]=zF(z)\big|_{z=0}=\left.\frac{z(z^2+2z+1)}{z(z^2-z+0.3561)}\right|_{z=0}$$

$$=1/0.3561=2.8082$$

$$\mathrm{Res}[F(z),p_1]=(z-p_1)F(z)\big|_{z=p_1}$$

$$=\frac{\cancel{(z-p_1)}(z^2+2z+1)}{z\cancel{(z-p_1)}(z-p_2)}$$

$$=\frac{(r\mathrm{e}^{\mathrm{j}\theta})^2+2r\mathrm{e}^{\mathrm{j}\theta}+1}{r\mathrm{e}^{\mathrm{j}\theta}(r\mathrm{e}^{\mathrm{j}\theta}-r\mathrm{e}^{-\mathrm{j}\theta})}$$

其中 $r=0.5967$，$\theta=33.08°$。注意到这个表达式与 4.24 式相同，我们可以重写为

$$\text{Res}[F(z), p_1] = -0.9041 - 5.9928\text{j}$$

由于 p_1、p_2 是复共轭对，那么

$$\text{Res}[F(z), p_2] = -0.9041 + 5.9928\text{j}$$

因此

$$\begin{aligned} x(0) &= \text{Res}[F(z), 0] + \text{Res}[F(z), p_1] + \text{Res}[F(z), p_2] \\ &= 2.8082 - 0.9041 - 5.9928\text{j} - 0.9041 + 5.9928\text{j} \\ &= 1 \end{aligned}$$

当 $n>0$ 时，在 $z=0$ 的极点消失，我们有

$$F(z) = \frac{z^n(z^2+2z+1)}{z(z^2-z+0.3561)}$$

$$x(n) = \text{Res}[F(z), p_1] + \text{Res}[F(z), p_2]$$

$$\begin{aligned} \text{Res}[F(z), p_1] &= (z-p_1)F(z)\big|_{z=p_1} \\ &= \left.\frac{\cancel{(z-p_1)}z^n(z^2+2z+1)}{z\cancel{(z-p_1)}(z-p_2)}\right|_{z=p_1} \\ &= \frac{(re^{j\theta})^n[(re^{j\theta})^2+2re^{j\theta}+1]}{re^{j\theta}(re^{j\theta}-re^{-j\theta})} \end{aligned} \tag{4.32}$$

其中 $r=0.5967$，$\theta=33.08°$。注意到这个表达式与 4.24 式相同，我们可以重写为

$$\begin{aligned} \text{Res}[F(z), p_1] &= (0.5967e^{j33.08})^n(6.060\,66e^{-j98.58}) \\ &= 6.060\,66(0.5967)^n[\cos(33.08n-98.58) \\ &\quad + \text{j}\sin(33.08n-98.58)] \end{aligned}$$

由于 p_2 和 p_1 是复共轭对，我们可以写成

$$\begin{aligned} \text{Res}[F(z), p_2] &= 6.060\,66(0.5967)^n[\cos(33.08n-98.58) \\ &\quad - \text{j}\sin(33.08n-98.58)] \end{aligned}$$

这样

$$\begin{aligned} x(n) &= \text{Res}[F(z), p_1] + \text{Res}[F(z), p_2] \\ &= 12.1213(0.5967)^n\cos(33.08n-98.58°), \quad n>0 \end{aligned}$$

用部分式展开法检查上面的结果。

例 4.9　$X(z)$包含一个二阶极点　求具有下列 z 变换的离散时间序列 $x(n)$：

$$X(z) = \frac{z^2}{(z-0.5)(z-1)^2}$$

解：

这个例子与用部分分式展开法的例 4.6 相同。根据留数法，离散时间为

$$x(n) = \sum_{k=1}^{M}\text{Res}[F(z), p_k]$$

其中

$$F(z)=z^{n-1}X(z)=\frac{z^{n+1}}{(z-0.5)(z-1)^2}$$

$F(z)$在$z=0.5$有一个简单极点，在$z=1$有一个二阶极点，因此，$x(n)$为

$$x(n)=\mathrm{Res}\,[F(z),p_1]+\mathrm{Res}\,[F(z),p_2]$$

$$\mathrm{Res}\,[F(z),0.5]=\frac{\cancel{(z-0.5)}z^{n+1}}{\cancel{(z-0.5)}(z-1)^2}=\left.\frac{z^{n+1}}{(z-1)^2}\right|_{z=0.5}$$

$$=0.5(0.5)^n/(0.5)^2=2(0.5)^n$$

$$\mathrm{Res}\,[F(z),1]=\frac{\mathrm{d}}{\mathrm{d}z}\left[\frac{\cancel{(z-1)^2}z^{n+1}}{(z-0.5)\cancel{(z-1)^2}}\right]$$

$$=\left.\frac{(z-0.5)(n+1)z^n-z^{n+1}}{(z-0.5)^2}\right|_{z=1}$$

$$=[(0.5)(n+1)-1]/(0.5)^2=2(n-1)$$

组合结果，我们有

$$x(n)=2[(n-1)+(0.5)^n]$$

上式与部分分式展开法得到的结果相同。

大家可能已经注意到部分分式展开法和留数法是有联系的，两种方法都要求计算留数，尽管方法不同。部分分式展开法要求计算$X(z)$的留数C_k，而留数法要求计算$z^{n-1}X(z)$的留数。当$X(z)$有一阶极点时，我们有

$$\mathrm{Res}\,[z^{n-1}X(z),p_k]=z^n\,\mathrm{Res}\,[X(z),p_k]=z^nC_k \tag{4.33}$$

因此，附录中给出的部分分式展开法的C语言程序也可以用在留数法中计算留数。

4.3.4 z反变换方法的比较

我们已经详细讨论了求z反变换的三种方法，即幂级数法、部分分式展开法和留数法。幂级数法的限制是它不能得出闭合形式的解（尽管在简单情况下可以推导出），但它是一种简单的方法，有助于计算机的实现。然而，由于递推性质，当z反变换的数据点数比较大的时候，要仔细考虑尽可能地使计算的数值误差达到最小，例如可以采用双精度计算。

部分分式展开法和留数法得出闭合形式的解，这两种方法的主要不足是需要对分母多项式进行因式分解，即求$X(z)$的极点。如果$X(z)$的阶数很高时，倘若不进行因式分解，求极点是件很困难的任务，这一主题将在4.5.1节中进一步讨论。如果$X(z)$包含高阶极点，两种方法可能也包含高阶微分。很显然，如果要求闭合形式的解，那么部分分式法和留数法是最合适的。部分分式法产生数字滤波器并行结构的系数时特别有用（参见4.5.11节）。留数法广泛应用于离散时间系统的量化误差分析（参见第13章）。

在本书中利用适当的工具如MATLAB或C语言程序大大简化了z变换和z反变换，在附录中给出了几个应用的例子。

4.4 z变换的性质

下面简要介绍在DSP中广泛应用的z变换的性质，这些性质的证明将在本章的习题中给出。

(1) **线性特性**　如果序列 $x_1(n)$、$x_2(n)$有 z 变换 $X_1(z)$和 $X_2(z)$，那么序列的线性组合的 z 变换为

$$ax_1(n)+bx_2(n)\rightarrow aX_1(z)+bX_2(z) \tag{4.34}$$

(2) **延迟或平移**　如果序列 $x(n)$的 z 变换为 $X(z)$，那么序列延迟 m 个抽样后的 z 变换为 $z^{-m}X(z)$。这一性质广泛应用于将离散时间系统的z变换函数转换成时域差分方程，反过来也一样，请参见4.5.8 节。

$$x(n)\rightarrow X(z)$$

$$x(n-m)\rightarrow z^{-m}X(z)$$

(3) **卷积**　给定离散时间 LTI 系统，输入为 $x(n)$，系统的冲激响应为 $h(k)$，系统的输出为

$$y(n)=\sum_{k=-\infty}^{\infty}h(k)x(n-k) \tag{4.35a}$$

根据 z 变换，输入和输出之间的关系为

$$Y(z)=H(z)X(z) \tag{4.36b}$$

其中 $X(z)$、$H(z)$和 $Y(z)$分别是 $x(n)$、$h(k)$和 $y(n)$的 z 变换。给定 $X(z)$和 $H(z)$，输出 $y(n)$可以通过 z 反变换 $Y(z)$得到。

可以看出，在4.25a式中的卷积运算已经变成了 z 域的乘法运算，$H(z)$也常常称为系统的传递函数。

(4) **微分**　如果 $X(z)$是 $x(n)$的 z 变换，那么 $nx(n)$的 z 变换可以通过求 $X(z)$的微分得到：

$$x(n)\rightarrow X(z)$$

$$nx(n)\rightarrow -z\frac{\mathrm{d}X(z)}{\mathrm{d}z} \tag{4.36}$$

当 $X(z)$包含多阶极点的时候，在求 z 的反变换时这一性质很有用。

(5) **与拉普拉斯变换的关系**　连续时间系统或信号通常用拉普拉斯变换描述。如果我们令 $z=\mathrm{e}^{sT}$，其中 s 是复拉普拉斯变量，

$$s=d+\mathrm{j}\omega$$

那么，

$$z=\mathrm{e}^{(d+\mathrm{j}\omega)T}=\mathrm{e}^{dT}\mathrm{e}^{\mathrm{j}\omega T} \tag{4.37}$$

这样，

$$|z|=\mathrm{e}^{dT}\text{ and }\angle z=\omega T=2\pi f/F_s=2\pi\omega/\omega_s$$

其中 ω_s（弧度/秒）是抽样频率，当 ω从 $-\infty$到 ∞变化时，s平面就映射到z平面，如图4.5所示。在s平面的整个jω轴映射到z平面的单位圆，s平面的左边映射到单位圆内部，右边映射到单位圆的外部。

根据频率响应，jω轴在 s 平面是最重要的。在这种情况下，$d=0$，在 s 平面的频率点与 z 平面的单位圆通过

$$z=\mathrm{e}^{\mathrm{j}\omega T} \tag{4.38}$$

相联系。表4.2给出了某些特殊频率如何从s平面映射到z平面。很显然，映射不是惟一的，例如，在s平面的两个频率$\omega=\omega_s$、$\omega=2\omega_s$映射到单位圆的同一个点上。

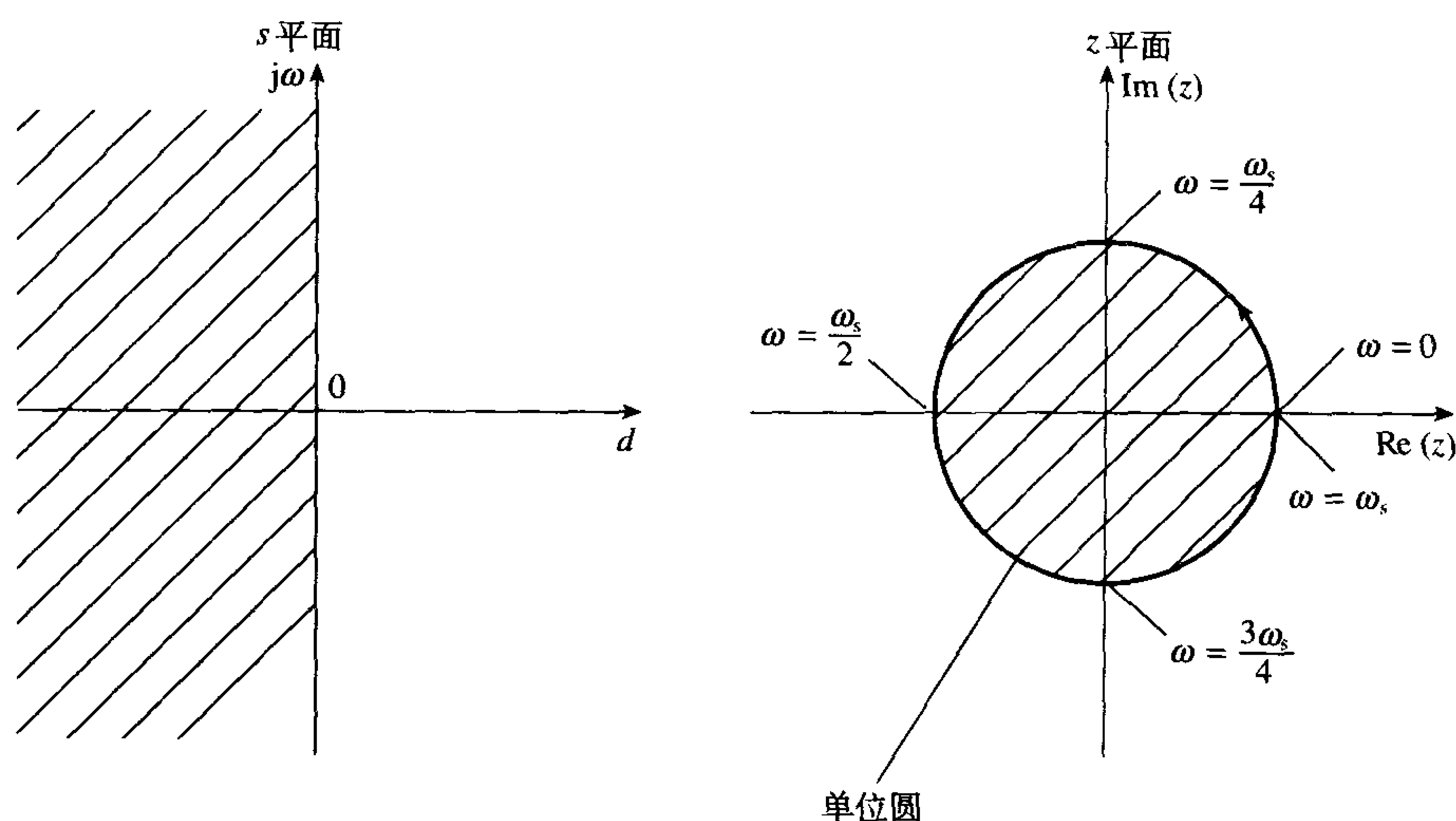


图4.5　s平面到z平面的映射，s平面的左边映射到z平面的单位圆内，右边映射到单位圆的外部，jω轴映射到单位圆

表4.2　从s平面的频率映射到z平面

s平面：ω（弧度/秒）	z平面：ωT（弧度）
0	0
$\omega_s/4$	$\pi/2$
$\omega_s/2$	π
$3\omega_s/4$	1.25π
ω_s	2π
$1.25\omega_s$	$\pi/2$
$1.5\omega_s$	π
$1.75\omega_s$	1.25π
$2\omega_s$	2π

4.5　z变换在信号处理中的应用

z变换在DSP系统中的应用有很多，许多应用将在后面的章节、特别是在第8章中详细讨论。下面几节重点介绍其中的一些应用，并建立一些共性的基本问题。

4.5.1　离散时间系统的极－零点描述

在大多数离散时间系统中，z变换即系统的传递函数可以用它的极点和零点来表示。例如，考虑下列z变换，它表示一般的N阶离散时间滤波器（其中$N=M$）：

$$H(z)=\frac{N(z)}{D(z)} \tag{4.39}$$

其中

$$N(z)=b_0z^N+b_1z^{N-1}+b_2z^{N-2}+\ldots+b_N$$
$$D(z)=a_0z^N+a_1z^{N-1}+a_2z^{N-2}+\ldots+a_N$$

a_k和b_k是滤波器的系数。

如果 $H(z)$有极点 $z = p_1, p_2,...,p_N$ 和零点 $z = z_1, z_2,...,z_N$，那么 $H(z)$可以因式分解为

$$H(z) = \frac{K(z - z_1)(z - z_2)\ldots(z - z_N)}{(z - p_1)(z - p_2)\ldots(z - p_N)} \tag{4.40}$$

其中 z_i 是第 i 个零点，p_i 是第 i 个极点，K 是增益因子。回想到 z 变换 $H(z)$的极点是使 $H(z)$变成无穷大的 z 值，使 $H(z)$变成零的 z 值称为零点。$H(z)$的极点和零点可能是实数或复数，当它们是复数时，它们是以共轭成对的形式出现的，以保证系数 a_k 和 b_k 是实数。由 4.40 式应该很清楚，如果 $H(z)$的极点和零点的位置是已知的，那么 $H(z)$自身是很容易重构的，只相差一个常数。

在 z 变换中所含的信息很容易用极零图显示出来，例如可以参见图 4.6。在这个图中，× 标出了极点的位置，○代表零点的位置。对于这个例子，极点在 $z = 0.5 \pm 0.5j$ 和 $z = 0.75$，单个零点在 $z = -1$。极零图一个很重要的特征是单位圆，即由 $|z|=1$ 定义的圆，参见图 4.6。后面将会很清楚地看到，单位圆在离散时间系统的分析与设计中起着重要的作用。

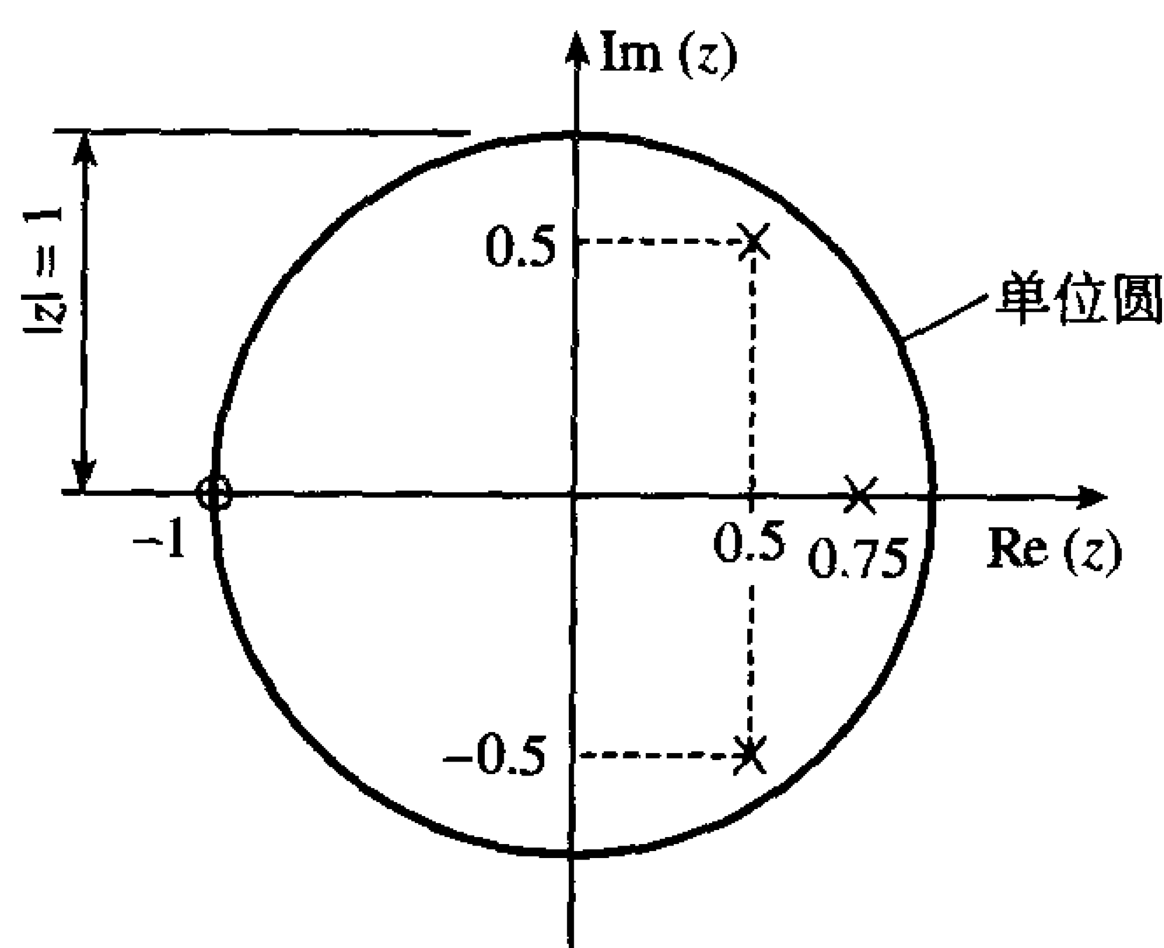


图 4.6　z 变换用极零图的形式描述：× 表示极点；○表示零点

极零图提供了给定离散时间系统的性质。例如，从极点和零点的位置我们可以推断系统的频率响应以及稳定度。对于一个稳定系统，所有极点都必须在单位圆内(或者与单位圆上的零点重合)。

通常，z 变换不具有可用的因式分解形式，而是具有像 4.39 那样的多项式之比的形式。在这些情况下，根据它的极点和零点描述 z 变换 $H(z)$，需要求分母多项式 $D(z)$和分子多项式 $N(z)$的根。

形式为 $ax^2 + bx + c$ 的二阶多项式的根为

$$\frac{-b \pm (b^2 - 4ac)^{1/2}}{2a} \tag{4.41}$$

对于高阶多项式，求 $N(z)$或 $D(z)$的根是个很困难的任务，在实际中常常是用数值计算的方法实现的。例如，牛顿（Newton）或 Baistow 算法（参见 Atkinson and Harley(1983)），在离散滤波器的设计和稳定性分析中常常需要求极点和零点。幸运的是，在设计离散时间滤波器时，极点和零点由滤波器设计软件自动产生，避免了直接求根的要求。

例 4.10

(1) 根据极点和零点表达下面的传递函数，并画出极零图：

$$H(z) = \frac{1 - z^{-1} - 2z^{-2}}{1 - 1.75z^{-1} + 1.25z^{-2} - 0.375z^{-3}}$$

(2) 离散滤波器的极零图如图 4.7 所示，求滤波器的传递函数 $H(z)$。

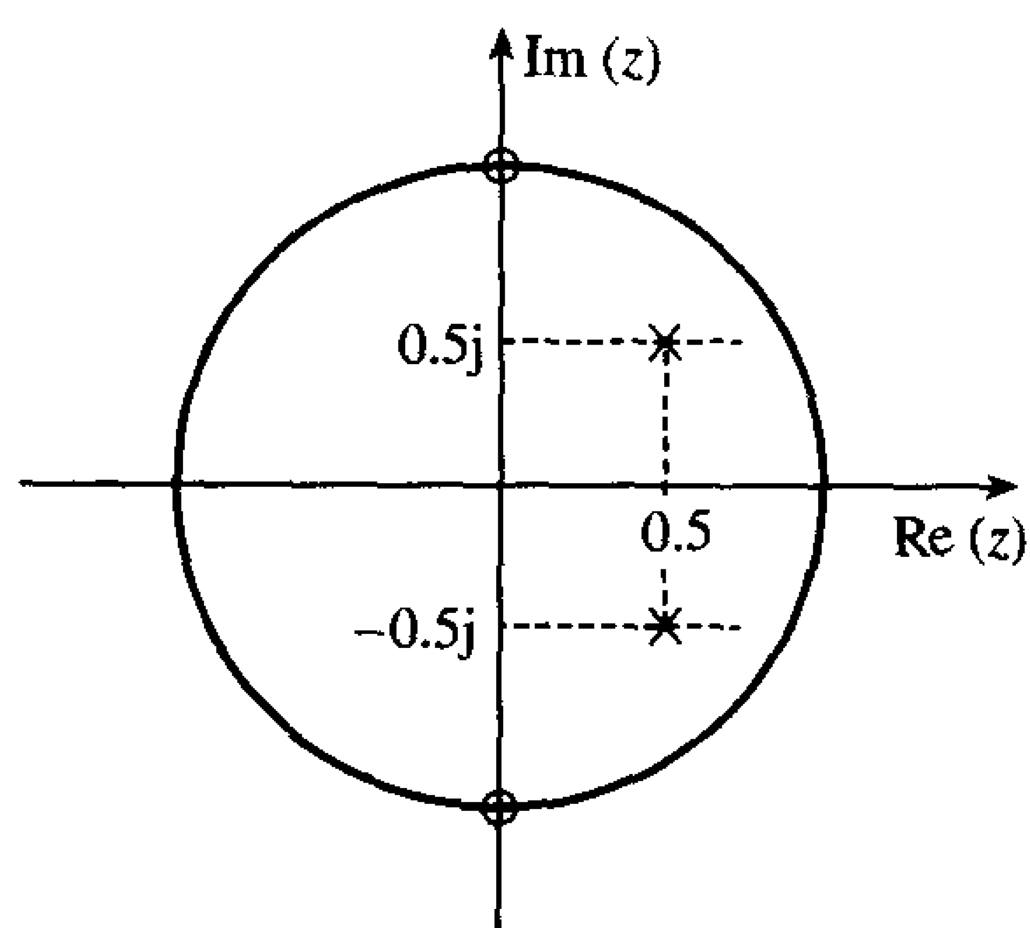


图 4.7 例 4.2 部分(2)的极零图

解：

(1) 首先，我们用 z 的正幂表示 $H(z)$，然后进行因式分解，求出极点和零点。在分子和分母同时乘以最高幂 z^3，可得

$$H(z)=\frac{z^3-z^2-2z}{z^3-1.75z^2+1.25z-0.375}$$

因式分解，得

$$H(z)=\frac{(z-2)(z+1)z}{(z-0.5+\mathrm{j}0.5)(z-0.5-\mathrm{j}0.5)(z-0.75)}$$

因此，极点位于 $z=0.5\pm0.5\mathrm{j}$ 和 $z=0.75$，零点位于 $z=2$、$z=-1$ 和 $z=0$，极零图如图 4.8 所示。

(2) 根据极零图，传递函数的零点在 $z=\pm\mathrm{j}$，极点在 $z=0.5\pm0.5\mathrm{j}$ 处。传递函数可直接写出为

$$\begin{aligned}H(z)&=\frac{K(z-\mathrm{j})(z+\mathrm{j})}{(z-0.5-0.5\mathrm{j})(z-0.5+0.5\mathrm{j})}\\&=\frac{K(z^2+1)}{z^2-z+0.5}\\&=\frac{K(1+z^{-2})}{1-z^{-1}-0.5z^{-2}}\end{aligned}$$

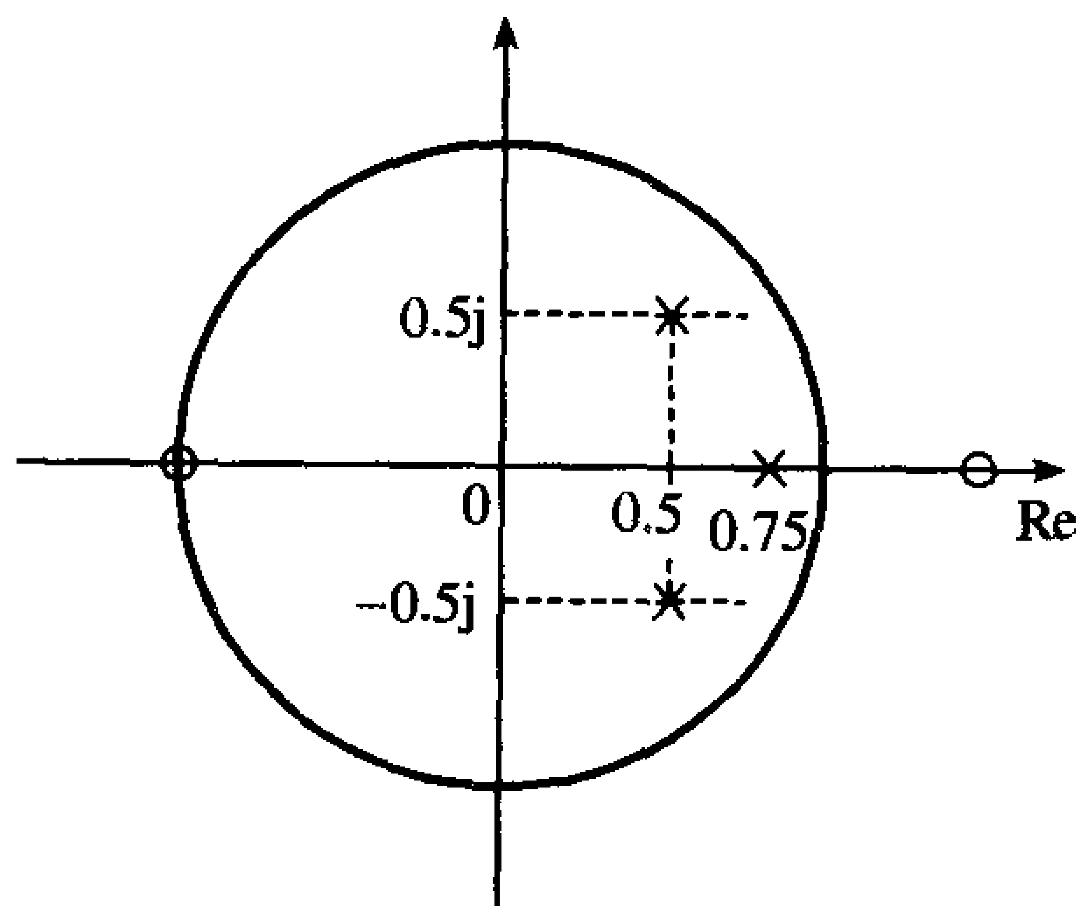


图 4.8 例 4.10 的极零图

4.5.2 频率响应估计

很多情况下需要计算离散时间系统的频率响应。例如，在离散滤波器的设计中，通常需要考察滤波器的频率，以便确保满足所要求的技术规范。系统的频率响应很容易从它的 z 变换得到。

例如，如果我们令 $z=e^{j\omega T}$，即计算单位圆上的 z 变换，那么我们就可以得到系统的傅里叶变换：

$$H(z)=\sum_{n=-\infty}^{\infty}h(n)z^{-n}\bigg|_{z=e^{j\omega T}} \tag{4.42a}$$

$$=H(e^{j\omega T})=\sum_{n=-\infty}^{\infty}h(n)e^{-jn\omega T} \tag{4.42b}$$

$H(e^{j\omega T})$称为系统的频率响应，我们用符号 T 强调离散时间系统的频率响应与抽样频率之间的关系。一般来说，$H(e^{j\omega T})$是复数，它的模给出了幅度响应，它的相位给出了系统的相位响应。

频率响应可以通过几种方法从 z 变换得到，我们将描述三种方法。

4.5.3　频率响应的几何计算

根据极零图求离散时间系统频率响应的大致形式，这是一种简单而又有用的方法。回忆一下 LTI 系统的 z 变换可以根据它的极零点表示为

$$H(z)=\frac{K(z-z_1)(z-z_2)\ldots(z-z_N)}{(z-p_1)(z-p_2)\ldots(z-p_N)}=\frac{\prod_{i=1}^{N}K(z-z_i)}{\prod_{i=1}^{N}(z-p_i)} \tag{4.33}$$

其中为了简单起见我们假定分子和分母的阶数是相等的，在 4.43 式中做替换 $z=e^{j\omega T}$，并且在区间（$0\leqslant\omega\leqslant\omega_s/2$）上计算 $H(e^{j\omega T})$可以求得频率响应。

$$H(e^{j\omega T})=\frac{\prod_{i=1}^{N}K(e^{j\omega T}-z_i)}{\prod_{i=1}^{N}(e^{j\omega T}-p_i)} \tag{4.44}$$

4.44 式当 z 变换只有两个零点和两个极点的几何解释在图 4.9 中给出。在这种情况下，频率响应为

$$H(e^{j\omega T})=\frac{K(e^{j\omega T}-z_1)(e^{j\omega T}-z_2)}{(e^{j\omega T}-p_1)(e^{j\omega T}-p_2)}$$

$$=\frac{KU_1\angle\theta_1U_2\angle\theta_2}{V_1\angle\phi_1V_2\angle\phi_2} \tag{4.45}$$

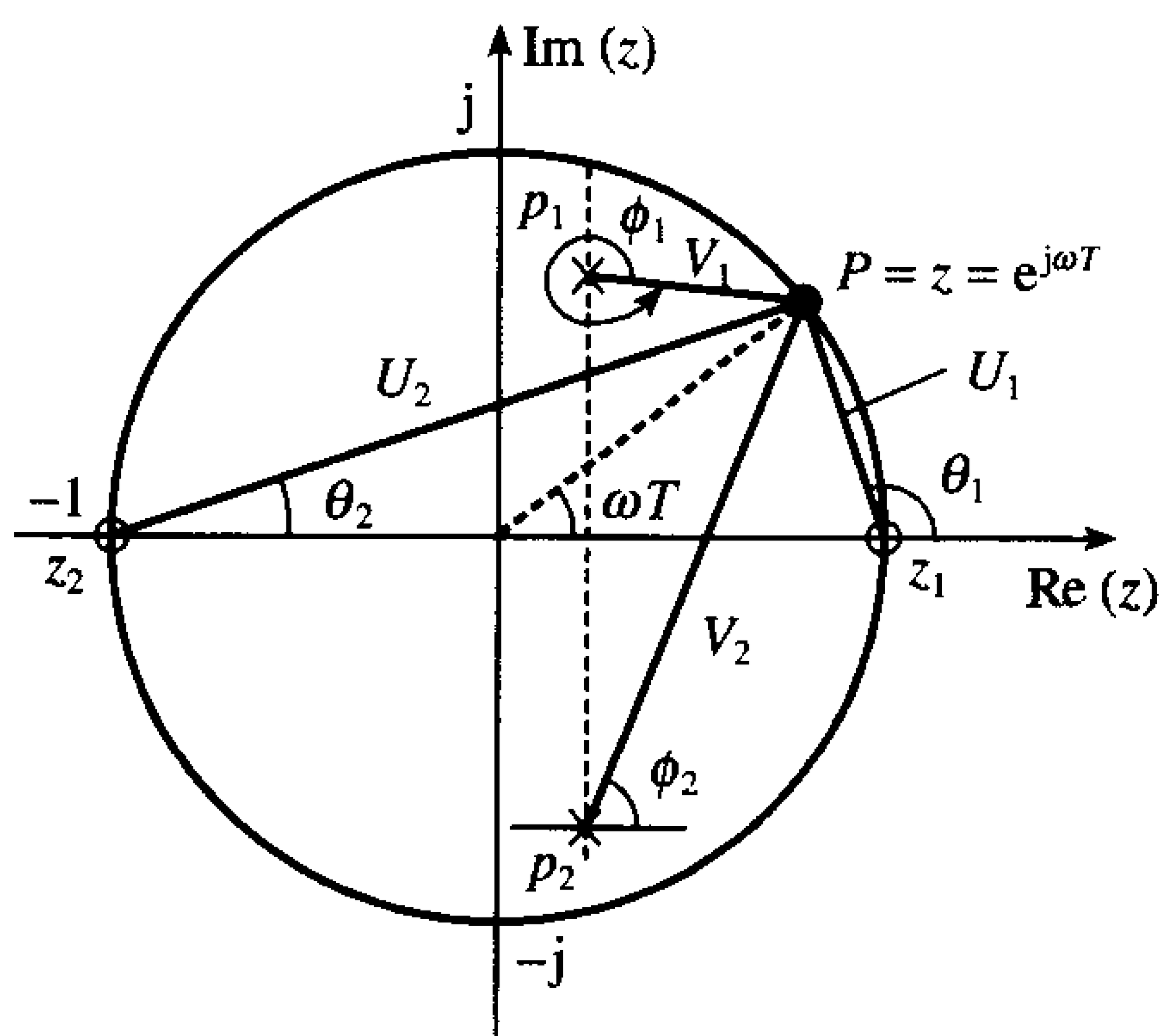


图 4.9　根据极零图的频率响应的几何计算

其中U_1和U_2表示从零点到点$z=e^{j\omega T}$的距离，V_1和V_2表示从极点到同一点的距离，如图4.9所示。因此，由4.45式，系统的幅度响应和相位响应为

$$|H(e^{j\omega T})| = \frac{U_1U_2}{V_1V_2},\ K=1$$

$$\angle[H(e^{j\omega T})] = \theta_1 + \theta_2 - (\phi_1 + \phi_2)$$

随着点P从$z=0$移到$z=-1$计算$H(e^{j\omega T})$就可以得到完整的频率响应。很明显，当点P移到靠近极点p_1时，矢量V_1的长度减小，所以幅度响应增加。另一方面，当点移到靠近零点z_1时，零点向量U_1减少，所以幅度响应$|H(e^{j\omega T})|$减小。这样，在极点处幅度响应出现峰值，而在零点处幅度响应降至零。

一般来说，利用几何方法，在某一给定频率点的频率响应是由零点矢量$U_i\angle\theta_i$（$i=1,2,\ldots$）的乘积和极点矢量$V_i\angle\phi_i$（$i=1,2,\ldots$）的乘积之比来确定。

例4.11 利用几何方法，求具有下列z变换的因果离散时间系统在dc、1/8、1/4、3/8和1/2抽样频率处的频率响应：

$$H(z) = \frac{z+1}{z-0.7071}$$

画出在间隔（$0\leqslant\omega\leqslant\omega_s$）上的幅度响应，其中$\omega_s$（弧度/秒）是抽样频率。

解：

在本例中，$H(z)$有一个单极点和单零点，如图4.10(a)的极零图所示。由4.44式，ω处的频率响应为

$$H(e^{j\omega T}) = \frac{U\angle\theta}{V\angle\phi} = \frac{e^{j\omega T}+1}{e^{j\omega T}-0.7071} = \frac{1+\cos(\omega T)+j\sin(\omega T)}{\cos(\omega T)-0.7071+j\sin(\omega T)} \tag{4.46}$$

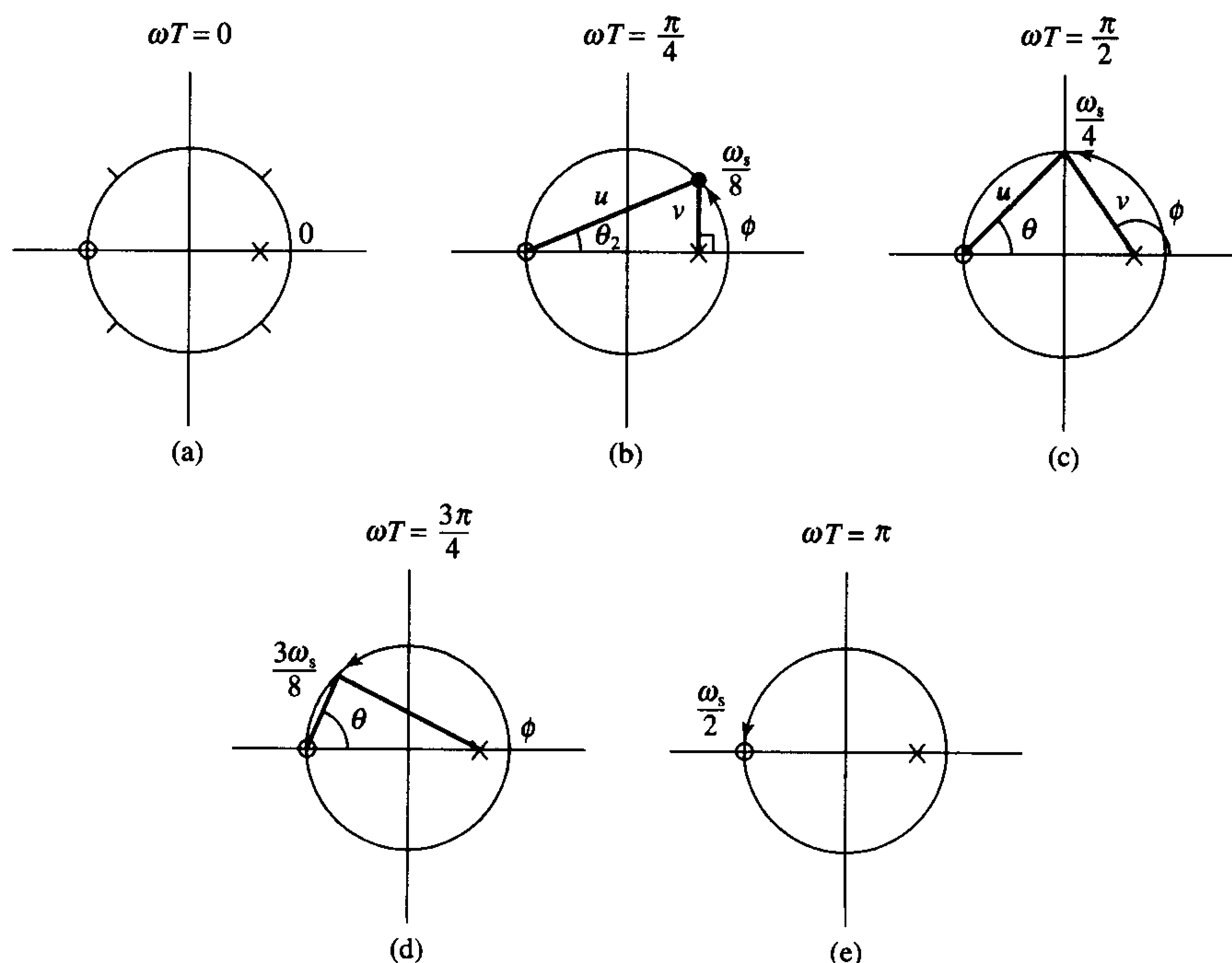


图4.10 采用几何方法和极零图的频率响应特性估计

在 dc 处，$\omega T=0$，零点矢量和极点矢量到点 $z=0$ 是 $2\angle 0°$ 和 $0.2929\angle 0°$，因此，频率响应为

$$H(e^{j\omega T})=2/0.2929=6.828\angle 0°$$

在 $\omega=\omega_s/8$ 处，$\omega T=\omega_s/8F_s=\pi/4$。在这种情况下，极点矢量和零点矢量如图 4.10(b)所示。我们将使用 4.46 右边的精确的表达式，这要胜于实际地测量矢量的长度和角度，这样，

$$H(e^{j\omega T})=\frac{1+\cos(\pi/4)+j\sin(\pi/4)}{\cos(\pi/4)-0.7071+j\sin(\pi/4)}$$

$$=\frac{1.8477\angle 22.5°}{0.7071\angle 90°}=2.6131\angle -67.5°$$

利用类似的方法，我们可以求得剩余的频率响应，总结如下，并在图 4.10(c) ~ 图 4.10(e)给出了矢量图。

ω(弧度/秒)	ωT（弧度）	$\lvert H(e^{j\omega T})\rvert$	$\angle H(e^{j\omega T})$（度）
0	0	6.828	0
$\omega_s/8$	$\pi/4$	2.6131	−67.5
$\omega_s/4$	$\pi/2$	1.1547	−80.26
$3\omega_s/8$	$3\pi/4$	0.4840	−85.93
$\omega_s/2$	π	0	0

幅度和相位响应如图4.11所示，需要注意的重要一点是，幅度响应$\lvert H(e^{j\omega T})\rvert$关于半抽样频率（奈奎斯特频率）是对称的，而相位响应是关于同一频率是反对称的，当离散时间系统的系数（a_k 和 b_k）是实数时总是这种情况。此外，这种系统的频率响应是周期的，周期为 ω_s（抽样频率），这一性质与抽样定理是一致的。

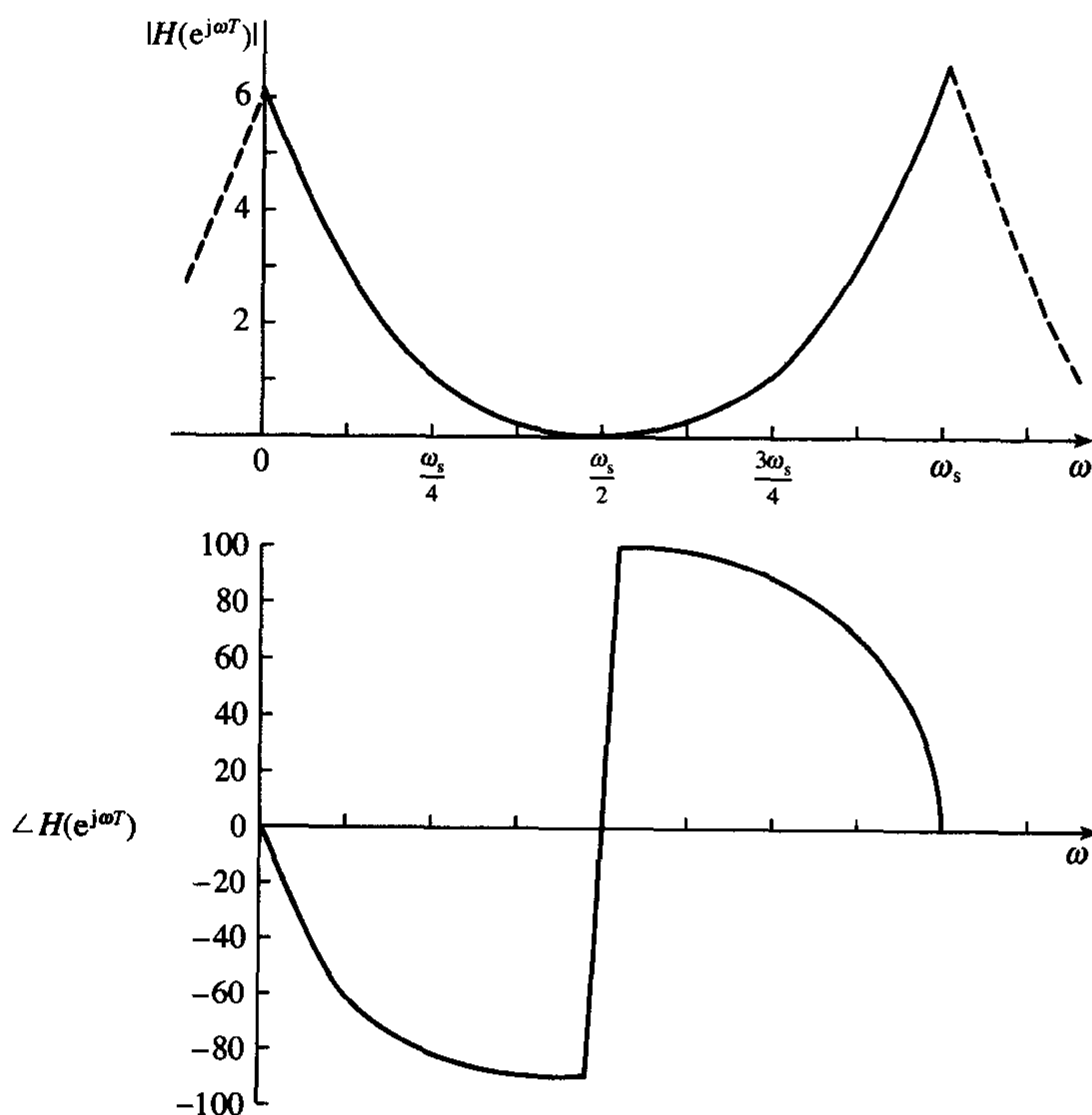


图 4.11　例 4.11 的离散时间的频率响应图

4.5.4 频率响应的直接计算机计算

频率响应的几何计算给出了频率响应的大致形状，但是如果要求对许多频率给出精确的响应是一件十分繁琐的事情。尽管可以自动处理，但是求极点和零点的位置的困难限制了它的应用。如果要求完整的频率响应，通常是直接把 $z = e^{j\omega T}$ 代入传递函数，然后计算导出的表达式：

$$H(e^{j\omega T}) = \left.\frac{b_0 + b_1 z^{-1} + \ldots + b_N z^{-N}}{a_0 + a_1 z^{-1} + \ldots + a_M z^{-M}}\right|_{z=e^{j\omega T}} \tag{4.47}$$

$$= \frac{b_0 + b_1 e^{-j\omega T} + \ldots + b_N e^{-jN\omega T}}{a_0 + a_1 e^{-j\omega T} + \ldots + a_M e^{-jM\omega T}}$$

$$= \frac{b_0 + b_1[\cos(\omega T) - j\sin(\omega T)] + \ldots + b_N[\cos(N\omega T) - j\sin(N\omega T)]}{a_0 + a_1[\cos(\omega T) - j\sin(\omega T)] + \ldots + a_M[\cos(M\omega T) - j\sin(M\omega T)]} \tag{4.48}$$

附录4C讨论了4.48式的C语言实现，程序在间隔 $0 \leqslant \omega \leqslant \omega_s/2$ 上计算 $H(e^{j\omega T})$。在附录4D描述了用MATLAB计算频率响应，并且给出了一个说明例子。

4.5.5 用FFT估计频率响应

FFT可以用来计算离散时间系统的频率响应。对于IIR系统，这样做的方法是首先求系统的冲激响应，例如用幂级数方法，然后计算冲激响应的FFT，这可以直接由4.42b式得到。4.42b式表明离散时间系统的频率响应简单地是冲激响应的傅里叶变换。为了得到平滑的频率响应，取足够多的冲激响应值是很重要的，或者在做FFT之前对冲激响应补零。在附录中讨论了C语言和MATLAB的实现。

另一种技术是首先对分子和分母系数补零，例如

$$\{b(n)\} = \{b_0, b_1, b_2, \ldots, b_M, 0, 0, \ldots, 0\}$$
$$\{a(n)\} = \{a_0, a_1, a_2, \ldots, a_N, 0, 0, \ldots, 0\} \tag{4.49}$$

然后分别求 $\{a(n)\}$ 和 $\{b(n)\}$ 的傅里叶变换 $A(k)$ 和 $B(k)$，两个FFT之比得到频率响应：

$$H(e^{j\omega_k T}) = A(k)/B(k), \quad k = 0, 1, \ldots, N/2 \tag{4.50}$$

4.5.6 在离散时间系统中使用的频率单位

连续时间系统和信号通常用拉普拉斯变换描述。因此，连续时间系统的频率响应通常是通过在系统的传递函数 $H(s)$ 中令 $s = j\omega$，其中 s 是复拉普拉斯变量。在DSP中，我们处理的是离散时间系统和信号，在这种情况下，通过令 $z = e^{j\omega T}$、然后在间隔 $0 \leqslant \omega \leqslant \omega_s/2$ 上计算 z 变换 $H(z)$ 来求得频率响应。在离散系统中的关键点是有用的频率范围依赖于抽样频率 ω_s。

表4.3说明了当 ω 从0变到 ω_s 时 ωT 和 z 是如何变化的。可以推断，当角度 ωT 从0到 2π 变化时，z 的值从1变到j、再回到1，如图4.12所示。从图中也可以很明显地看出，离散时间系统的频率响应是周期的：当我们绕圆走一圈或几圈时，z 的值是简单地重复。

通常用两个频率单位来描述离散时间系统的频率响应，即 ω（弧度/秒）和 f（Hz）。当频率单位为弧度/秒时，频率响应从 $\omega = 0$ 到 $\omega = \omega_s/2$，或等价于从 $\omega = 0$ 到 $\omega = \pi/T$（因为 $\omega_s = 2\pi F_s = 2\pi/T$）。当用标准的频率单位赫兹时，频率范围从0到 $F_s/2$，或等价于从0到 $1/2T$。两种频率单位也可以用归一化的形式表示，即 $T = 1$，或等价于 $F_s = 1$。表4.3说明了两个频率单位之间的关系，因此，感兴趣的频率范围可以用下列六种方式之一表示：

$$
\left.\begin{array}{ll}
0 \leqslant \omega \leqslant \omega_s/2 & \text{弧/秒} \\
0 \leqslant \omega \leqslant \pi/T & \text{弧/秒} \\
0 \leqslant \omega \leqslant \pi & \text{(归一化)}
\end{array}\right\} \tag{4.51}
$$

$$
\left.\begin{array}{ll}
0 \leqslant f \leqslant F_s/2 & \text{Hz} \\
0 \leqslant f \leqslant 1/2T & \text{Hz} \\
0 \leqslant f \leqslant 1/2 & \text{(归一化)}
\end{array}\right\} \tag{4.52}
$$

当我们在考察频率响应图或刻画离散时间系统时，Hz的单位更具有吸引力（且不会混淆）。然而，在DSP中遇到许多数学计算公式时，用弧度/秒更为方便。

表4.3　离散时间系统中使用的频率单位以及它们与单位圆上的点之间的关系

f(Hz)	ω（弧度/秒）	ωT（弧度）	$z=e^{j\omega T}$
0	0	0	1
$\frac{F_s}{8}$	$\frac{\omega_s}{8}$	$\frac{\pi}{4}$	$\frac{\sqrt{2}}{2}+\frac{\sqrt{2}}{2}j$
$\frac{F_s}{4}$	$\frac{\omega_s}{4}$	$\frac{\pi}{2}$	j
$\frac{3F_s}{8}$	$\frac{3\omega_s}{8}$	$\frac{3\pi}{4}$	$-\frac{\sqrt{2}}{2}+\frac{\sqrt{2}}{2}j$
$\frac{F_s}{2}$	$\frac{\omega_s}{2}$	π	−1
$\frac{5F_s}{8}$	$\frac{5\omega_s}{8}$	$\frac{5\pi}{4}$	$-\frac{\sqrt{2}}{2}-\frac{\sqrt{2}}{2}j$
$\frac{3F_s}{4}$	$\frac{3\omega_s}{4}$	$\frac{3\pi}{2}$	−j
$\frac{7F_s}{8}$	$\frac{7\omega_s}{8}$	$\frac{7\pi}{4}$	$\frac{\sqrt{2}}{2}-\frac{\sqrt{2}}{2}j$
F_s	ω_s	2π	1

$F_s=1/T$是抽样频率，单位为Hz；T是抽样周期；$\omega_s=2\pi/T$是抽样频率，单位为弧度/秒。

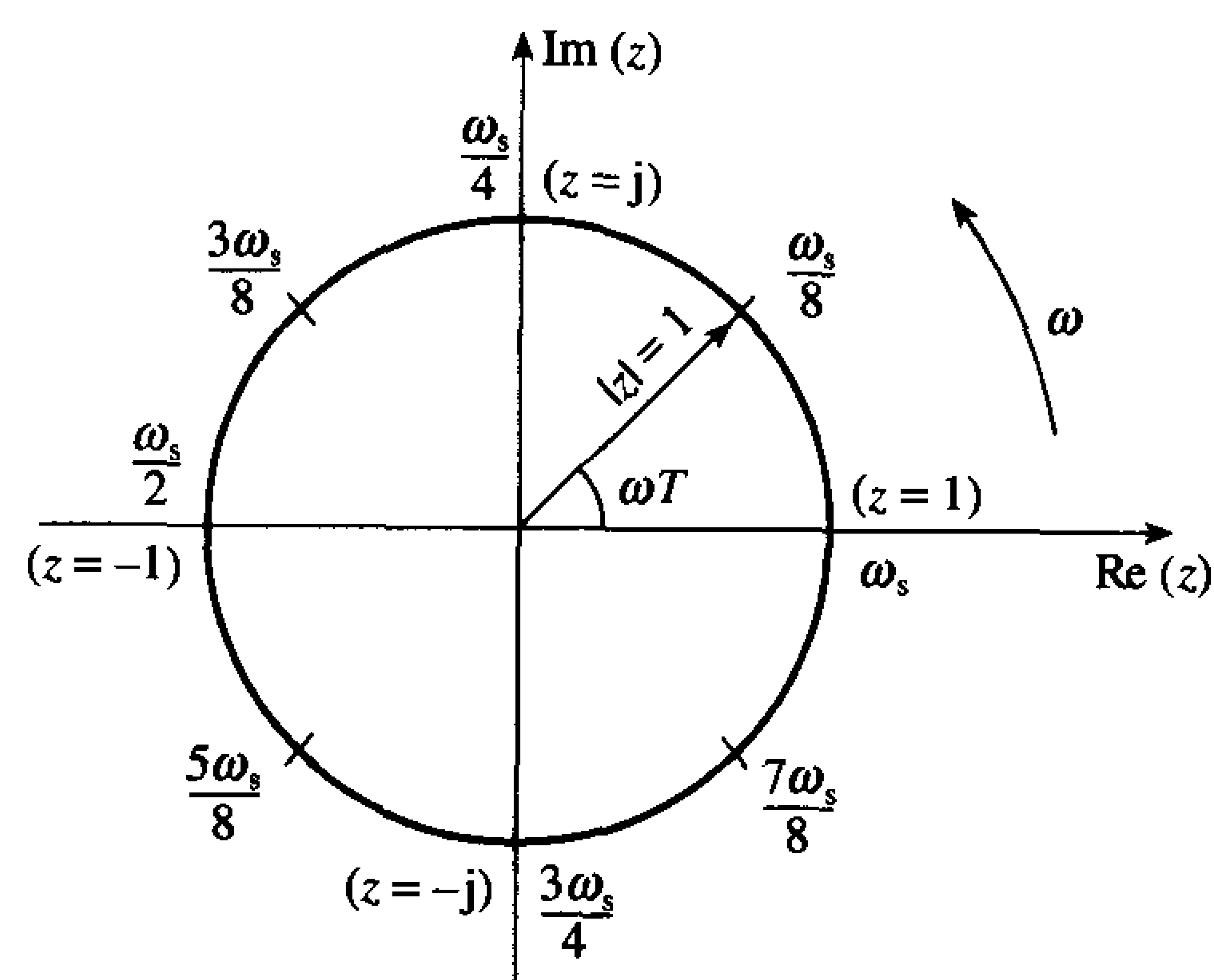


图4.12　z平面单位圆上的关键频率点

例4.12　给定一带通滤波器的频率响应指标（用Hz表示）如下：

通带	6 ~ 10 kHz
阻带	0 ~ 4 kHz和12 ~ 16 kHz
抽样频率	32 kHz

(1) 用归一化频率f表示指标；

(2) 将指标由标准的 Hz 单位转换为弧度/秒；

(3) 将部分(2)用弧度/秒表示的指标转换成用标准频率 ω 表示。

解：

(1) 通过用抽样频率除以每个频率，Hz 表示的带沿频率就可以用归一化的形式来表示。因此，归一化形式表示的指标变成

通带	0.1875 ~ 0.3125
阻带	0 ~ 0.125 和 0.375 ~ 0.5
抽样频率	1

(2) 由于 $\omega=2\pi f$，将每个带沿频率简单地乘以 2π 就可以将它转换成弧度/秒，这时，频率响应指标变成

通带	$12\,000\pi$ ~ $20\,000\pi$ 弧度/秒
阻带	0 ~ 8000π 和 $24\,000\pi$ ~ $32\,000\pi$ 弧度/秒
抽样频率	$64\,000\pi$ 弧度/秒

(3) 在(2) 中的带沿频率通过对每个频率除以 32 kHz（抽样频率）就可以用归一化形式表示，例如

$$12\,000\pi \to \frac{12\,000\pi}{32\,000}=\frac{3\pi}{8}$$

这样，指标变成

通带	$3\pi/8$ ~ $5\pi/8$
阻带	0 ~ $\pi/4$ 和 $3\pi/4$ ~ π
抽样频率	2π

4.5.7 稳定性考虑

稳定性分析常常作为离散系统设计的一部分来进行。一个有用的LSI系统稳定性准则是所有有界的输入产生有界的输出。这就是所谓的 BIBO（输入有界，输出有界）条件。说明 LSI 系统是稳定的，当且仅当它满足准则：

$$\sum_{k=0}^{\infty}|h(k)|<\infty \tag{4.53}$$

其中$h(k)$是系统的冲激响应。很显然，如果冲激响应是有限长度的，由于冲激响应系数之和是有限的，上述条件满足。因此，稳定性的考虑只是应用到具有无限持续时间冲激响应的系统。

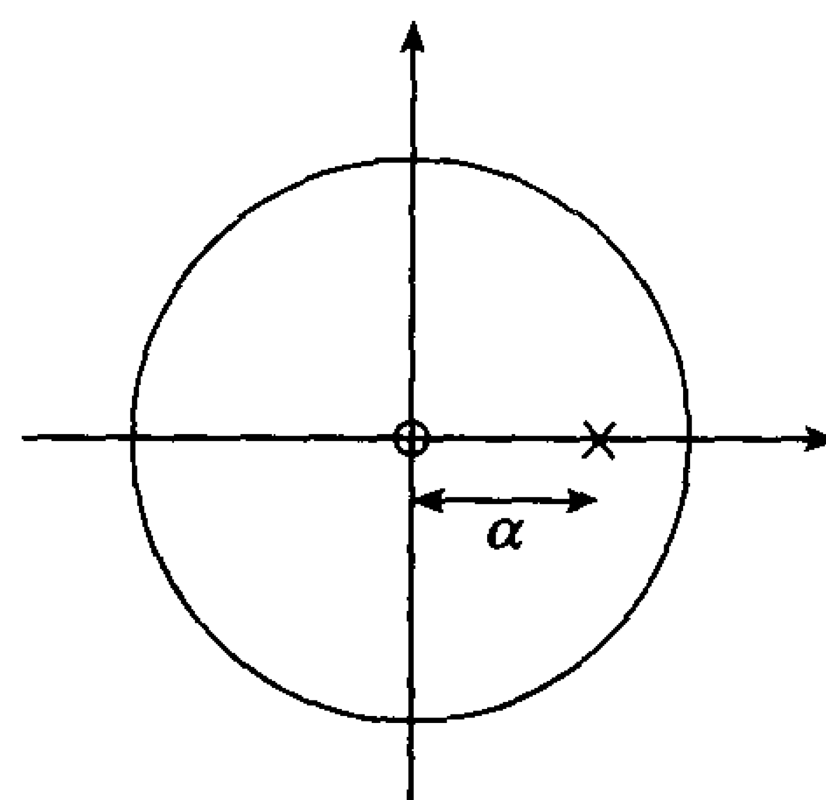
对于有界的输出，所有极点必须位于单位圆内。当极点位于单位圆外时，系统是不稳定的。在实际中，在单位圆上有极点的系统也认为是不稳定的或者是潜在不稳定的，因为微小的扰动或误差必定使系统进入到不稳定态，一个例外是单位圆上的极点与零点重合。对于不稳定的系统，冲激响应随时间增大。

从原理上讲，稳定性的测试是简单的：找出z变换极点的位置，如果在圆上或圆外存在任何极点（除非在单位圆上与零点重合），那么系统是不稳定的。在实际中，求极点的位置可能并不是一件容易的事。

当系统的 z 变换 $H(z)$ 不能进行因式分解时，一种可能的简单测试方法是通过求 z 反变换来求出和画出足够多的冲激响应值。如果冲激响应值随时间无限增大，或者没有足够快地衰减，那么系统是不稳定的或者是临界稳定的。图4.13给出了一个简单离散时间系统对于不同稳定度冲激响应特性的例子，其他更精细的稳定性测试方法在一些有关 z 变换的高级教程中可以找到（例如Jury, 1964; Proakis and Manolakis, 1992）。对于二阶系统的稳定性讨论将在第 8 章中给出。

	$h(n)$			
n	$\alpha = 0.5$	$\alpha = 0.99$	$\alpha = 1$	$\alpha = 1.5$
0	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00
1	1.00000e+01	1.00000e+01	1.00000e+01	1.00000e+01
2	5.00000e+00	9.90000e+00	1.00000e+01	1.50000e+01
3	2.50000e+00	9.80100e+00	1.00000e+01	2.25000e+01
4	1.25000e+00	9.70299e+00	1.00000e+01	3.37500e+01
5	6.25000e-01	9.60596e+00	1.00000e+01	5.06250e+01
6	3.12500e-01	9.50990e+00	1.00000e+01	7.59375e+01
7	1.56250e-01	9.41480e+00	1.00000e+01	1.13906e+02
8	7.81250e-02	9.32065e+00	1.00000e+01	1.70859e+02
9	3.90625e-02	9.22745e+00	1.00000e+01	2.56289e+02

(a)



(b)

图 4.13　对于不同的稳定度，系统冲激响应特性的说明：(a) 冲激响应；(b) z 平面极零图。系统的 z 变换是 $10z^{-1}/(1-\alpha z^{-1})$。(i)当 $\alpha = 0.5$ 时系统是稳定的，(ii)当 $\alpha = 0.99$ 时，系统是临界稳定的，(iii)当 $\alpha = 1$ 时，系统是潜在不稳定的，(iv)当 $\alpha = 1.5$ 时，系统是不稳定的。例如，当 $\alpha = 0.5$ 时，冲激响应值随 n 迅速衰减；而当 $\alpha = 1.5$ 时，冲激响应值迅速增加

4.5.8　差分方程

差分方程在时域刻画了离散时间系统为了产生期望的输出对输入数据必须执行的实际运算。对于大多数感兴趣的实际情况，差分方程可以写成

$$y(n) = \sum_{k=0}^{N} a_k x(n-k) - \sum_{k=1}^{M} b_k y(n-k) \tag{4.54}$$

其中 $x(n)$ 是输入抽样值，$y(n)$ 是输出抽样值，$y(n-k)$ 是前面的输出，a_k 和 b_k 是系统的系数。4.54式表明当前的输出 $y(n)$ 是从当前和过去的输入抽样值以及以前的输出 $y(n-k)$ 得到的。

离散时间系统的差分方程很容易从它们的传递函数得到。利用 z 变换的性质也可以很容易地从传递函数得到系统的差分方程：

$$a_k x(n) \leftrightarrow a_k X(z)$$

$$a_k x(n-k) \leftrightarrow a_k z^{-k} X(z)$$

因此，4.54 式可以写成

$$Y(z) = \sum_{k=0}^{N} a_k z^{-k} X(z) - \sum_{k=0}^{M} b_k z^{-k} Y(z) \tag{4.55}$$

化简后我们得到离散系统的 z 域传递函数 $H(z)$ 为

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{k=0}^{N} a_k z^{-k} \Bigg/ \left(1 + \sum_{k=0}^{M} b_k z^{-k}\right) \tag{4.56}$$

如果所有分母的系数 b_k 是零，4.54 式和 4.55 式可化简为

$$y(n) = \sum_{k=0}^{N} a_k x(n-k) \tag{4.57a}$$

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{k=0}^{N} a_k z^{-k} \tag{4.57b}$$

系统的输出$y(n)$现在只与当前和过去的输入抽样值有关，而与4.54式那样的以前输出无关。在这种情况下，系数a_k表示系统的冲激响应，通常用符号$h(k)$表示。这类LTI系统称为有限冲激响应（FIR）系统，因为$h(k)$的长度是有限的。

4.54式和4.56式中的分母系数至少有一个系数不为零，它们所刻画的系统称为无限冲激响应（IIR）系统。在IIR系统中至少有一个极点非零，但在FIR系统中通常不含极点。

4.5.9 冲激响应估计

在离散时间的设计中，常常需要得到冲激响应值。例如，在FIR系统的设计中，为了实现系统要求冲激响应。在IIR系统的设计中，稳定性的分析也要求冲激响应的值。冲激响应也可以用来计算系统的频率响应。

离散时间系统的冲激响应可以定义为系统传递函数$H(z)$的z反变换：

$$h(k) = Z^{-1}[H(z)], \quad k = 0, 1, \ldots$$

如果z变换$H(z)$可以表示为幂级数形式，即

$$\begin{aligned} H(z) &= \sum_{n=0}^{\infty} h(n) z^{-n} \\ &= h(0) + h(1)z^{-1} + h(2)z^{-2} + \ldots \end{aligned} \tag{4.58}$$

那么z变换的系数就直接给出了冲激响应。对于IIR系统，$H(z)$常常表示为4.47式那样的多项式之比。在这种情况下，可以采用4.3节中描述的IZT方法来得到系统的冲激响应，附录中描述了用于此目的C语言和MATLAB程序。

冲激响应也可以看作为离散时间系统对单位冲激$u(n)$的响应。单位冲激信号$u(n)$当$n = 0$时为1，对其他n值为0。这种观点来自于这样的事实：如果我们加到系统的输入为单位冲激，即$x(n) = u(n)$，那么系统的输出等于$h(n)$，系统的冲激响应（严格地说是单位取值响应）为

$$\begin{aligned} y(n) &= \sum_{k=0}^{\infty} h(k)x(n-k) = \sum_{k=0}^{\infty} h(k)u(n-k) \\ &= h(0)u(n) + h(1)u(n-1) + h(2)u(n-2) + \ldots \\ &= h(n), \quad n = 0, 1, \ldots \end{aligned} \tag{4.59}$$

这提供了一种简单的可供选择的计算$h(n)$的方法（甚至它提供了求z反变换的另一种方法）。下面通过一个例子来加以说明。

例4.13 求由下列z变换刻画的离散时间滤波器的冲激响应，

(1) 用幂级数展开的方法

(2) 将单位冲激信号应用到系统：

$$H(z) = \frac{1 - z^{-1}}{1 + 0.5z^{-1}}$$

解：

(1) 用幂级数展开法，冲激响应值求得如下：

$$
\begin{array}{r l}
 & 1 - 1.5z^{-1} + 0.75z^{-2} - 0.375z^{-3} \ldots \\
1 + 0.5z^{-1} & \overline{)\,1 - z^{-1}} \\
 & \underline{1 + 0.5z^{-1}} \\
 & -1.5z^{-1} \\
 & \underline{-1.5z^{-1} - 0.75z^{-2}} \\
 & 0.75z^{-2} \\
 & \underline{0.75z^{-2} + 0.375z^{-3}} \\
 & -0.375z^{-3}
\end{array}
$$

从商可得到冲激响应值为

$$h(0) = 1, h(1) = -1.5, h(2) = 0.75, h(3) = -0.325$$

当然，冲激响应值也可以借助附录给出的幂级数的C语言程序来求得。

(2) 首先，我们需要从传递函数得到滤波器的差分方程：

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1 - z^{-1}}{1 + 0.5z^{-1}}$$

交叉相乘并利用z变换的延迟性质，我们得到如下差分方程：

$$Y(z) + 0.5Y(z)z^{-1} = X(z) - X(z)z^{-1}$$

$$y(n) + 0.5y(n-1) = x(n) - x(n-1)$$

化简得

$$y(n) = x(n) - x(n-1) - 0.5y(n-1)$$

而滤波器的冲激响应能够通过令$x(n) = u(n)$来得到，其中

$$u(n) = 1, \quad n = 0$$
$$= 0, \quad n \neq 0$$

且假定起始条件$y(-1) = 0$：

$$y(0) = 1$$

$$y(1) = x(1) - x(0) - 0.5y(0) = 0 - 1 - 0.5 = -1.5$$

$$y(2) = x(2) - x(1) - 0.5y(1) = -0.5 \times -1.5 = 0.75$$

$$y(3) = x(3) - x(2) - 0.5y(2) = -0.5 \times 0.75 = -0.325$$

$$\vdots$$

由此可得冲激响应值为

$$h(0) = 1, h(1) = -1.5, h(2) = 0.75, h(3) = -0.325$$

可以看出，两种方法得出了相同的结果。

4.5.10　在数字滤波器设计中的应用

z变换在DSP中的最重要的应用之一是数字滤波器的设计以及误差分析，特别是IIR滤波器。z变换广泛地用于确定数字滤波器的系数，并且分析量化误差对数字滤波器性能的影响。例如，众

所周知，当离散时间系统用硬件或软件实现时，由于实际处理器的寄存器字长有限，量化误差是固有的。z变换为分析这些误差对系统性能的影响提供了便利的工具。在实际中，由差分方程表示的乘法运算由于舍入或截断引起的误差常常借助z变换进行分析。离散时间滤波器的噪声分析将在第 13 章进行详细的讨论。

z变换在离散滤波器设计中的另一个重要应用是数字滤波器结构的表示。我们将在这里详细地进行讨论，因为它要应用前面提到的部分分式展开程序。

4.5.11 数字滤波器的实现结构

离散时间滤波器常常用框图或者信号流图的形式表示。框图或信号流图是表示差分方程或者等价地是表示传递函数的方便方法。例如，考虑一个具有下列差分方程的简单滤波器：

$$y(n) = x(n-1) - b_1 y(n-1) + b_2 y(n-2) + b_3 y(n-3) \tag{4.60}$$

这个方程的框图表示如图4.14(a)所示，图中符号z^{-1}代表一个单位延迟，这可以从各个节点的信号推断出来，箭头表示乘法器，靠近箭头的常数表示一个乘因子。差分方程与框图之间的关系应该是很明显的。同一个差分方程的信号流图表示如图4.14(b)所示。通常称框图和信号流图为实现图。

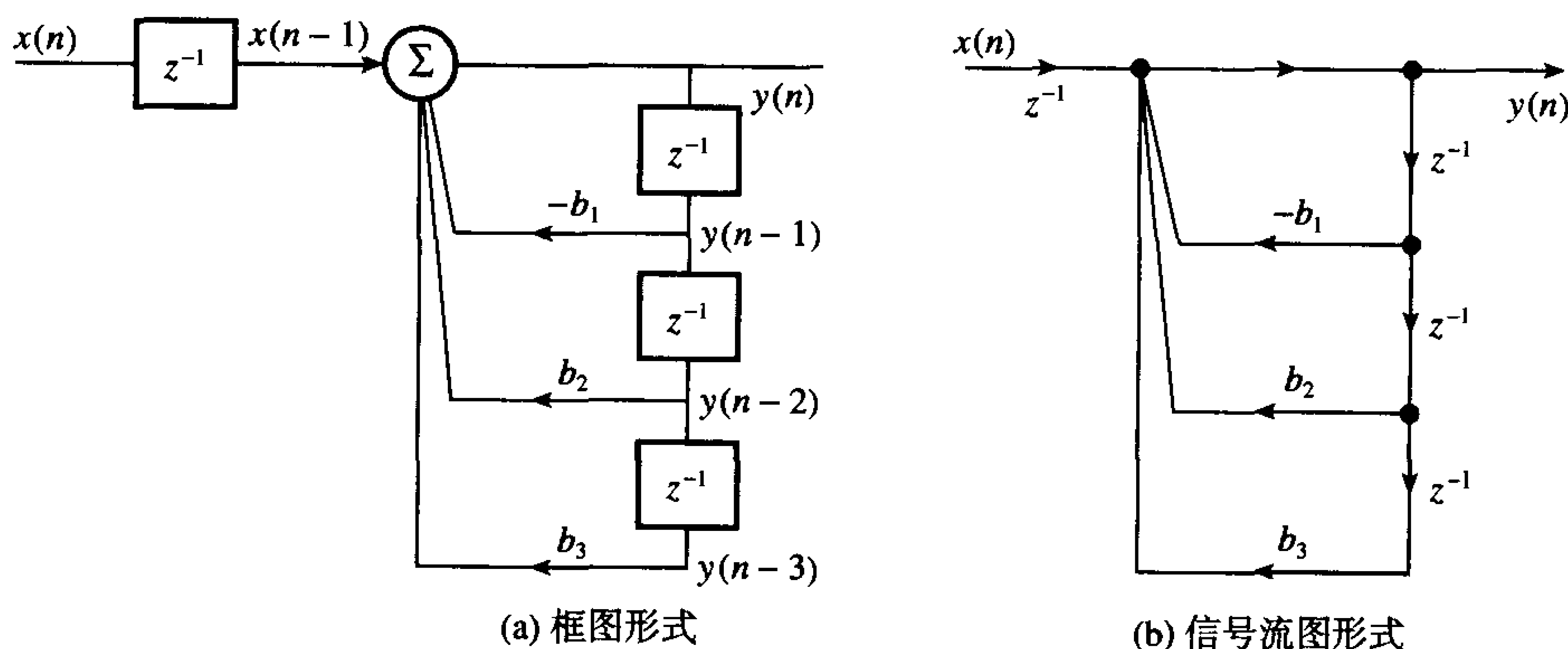


图 4.14 差分方程的实现图

当$H(z)$的阶数很高时，离散时间滤波器很少直接用图 4.14 的形式实现，因为如果用来表示系数并且执行差分方程的位数少将导致较大的误差（参见第8章和第13章）。实际中通常的方法是将传递函数分解成二阶或一阶z变换的串行或并行组合形式，对于串行实现，传递函数因式分解为

$$H(z) = H_1(z)H_2(z)\ldots H_K(z) = \prod_{i=1}^{K} H_i(z) \tag{4.61}$$

其中$H_i(z)$是二阶或一阶部分：

$$H_i(z) = \frac{b_0 + b_{1i}z^{-1} + b_{2i}z^{-2}}{1 + a_{1i}z^{-1} + a_{2i}z^{-2}} \qquad \text{二阶}$$

$$H_i(z) = \frac{b_0 + b_{1i}z^{-1}}{1 + a_{1i}z^{-1}} \qquad \text{一阶}$$

K是$(M+1)/2$的整数部分，整个z变换是各z变换之积：参见图 4.15。

对于并行实现，传递函数被分解，使用部分分式，得

$$H(z) = B_0 + \sum_{i=1}^{K} H_i(z) \tag{4.62}$$

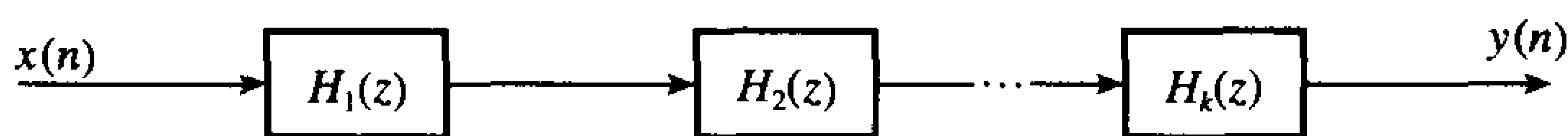


图4.15　串行实现的一般结构

其中$H_i(z)$是二阶或一阶部分，并且具有如下形式：

$$H_i(z) = \frac{a_{0i} + a_{1i}z^{-1}}{1 + b_{1i}z^{-1} + b_{2i}z^{-2}} \quad 二阶$$

$$H_i(z) = \frac{a_0}{1 + b_{1i}z^{-1}} \quad 一阶$$

其中K是$(M+1)/2$的整数部分，且

$$B_0 = a_N / b_M$$

图4.16画出了并行实现的一般结构。

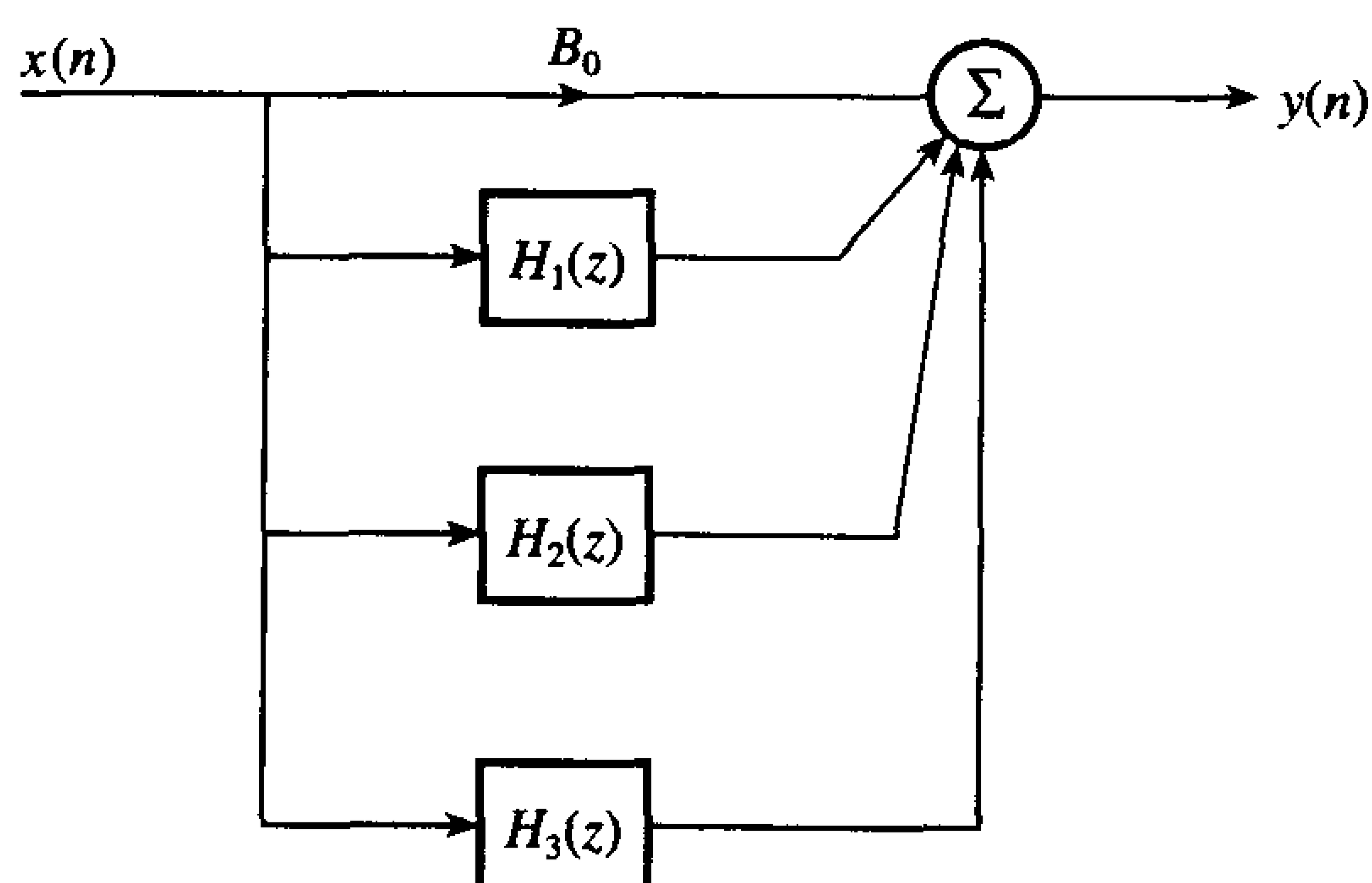


图4.16　并行实现的一般结构

在数字滤波器的设计中，软件包通常用来得到系数a_{ki}和b_{ki}。遗憾的是大多数软件包只生成串行结构的系数，并行结构的系数能够从串行结构通过部分分式展开而得到。我们通过一个例子加以说明。

例4.14　假定离散时间系统的z变换函数为

$$H(z) = \frac{1 - 2z^{-2} + z^{-4}}{1 - 0.414\,21z^{-1} + 0.085\,79z^{-2} + 0.292\,895z^{-3} + 0.5z^{-4}}$$

(1) 为了串行实现，利用二阶部分用适当形式表示$H(z)$。

(2) 为了并行实现，重复(1)。

解：

(1) 利用因式分解，$H(z)$表示为

$$H(z) = H_1(z)H_2(z)$$

其中

$$H_1(z) = \frac{1 - 2z^{-1} + z^{-2}}{1 - 1.414\,21z^{-1} + z^{-2}} \tag{4.63a}$$

$$H_2(z) = \frac{1 + 2z^{-1} + z^{-2}}{1 + z^{-1} + 0.5z^{-2}} \tag{4.63b}$$

(2) 为了并行实现，以适当的形式表示 $H(z)$，我们首先用部分分式展开的方法展开，这样

$$H(z) = B_0 + \frac{C_1}{z - p_1} + \frac{C_2}{z - p_2} + \frac{C_3}{z - p_3} + \frac{C_4}{z - p_4} \tag{4.64}$$

应用附录4B给出的PFE程序，极点 p_1 到 p_4 和系数 B_0、C_1 到 C_4 求得如下：

$p_1 = 0.7071 + 0.7071\text{j} = \text{e}^{\text{j}0.785}$; $\quad p_2 = p_1^*$

$p_3 = -0.5 + 0.5\text{j} = 0.7071\text{e}^{\text{j}2.356\,19}$; $\quad p_4 = p_3^*$

$B_0 = 2$

$C_1 = 0.114\,383 + 0.666\,669\text{j} = 0.676\,410\,4\angle 1.400\,877$; $\quad C_2 = C_1^*$

$C_3 = -0.614\,382\,76 - 0.580\,880\,79\text{j}$

$\quad = 0.845\,510\,897\,6\angle 3.898\,969$; $\quad C_4 = C_3^*$

其中角度是弧度。求出极点和系数 C_k、B_0 以后，4.64式的部分分式必须进行合并，使得 $H(z)$ 为二阶部分的和，即

$$H(z) = B_0 + \sum_{i=1}^{2} H_i(z) \tag{4.65a}$$

其中

$$H_i(z) = \frac{a_0 + a_{1i}z^{-1}}{1 + b_{1i}z^{-1} + b_{2i}z^{-2}} \tag{4.65b}$$

为了保证4.65b式中的系数 a_{ki} 和 b_{ki} 是实的，4.64式中含 C_1 和 C_2 的部分分式必须合并，因为它们是复共轭对。类似地，含 C_3 和 C_4 的部分分式也必须合并。合并含 C_1 和 C_2 的部分分式，我们得

$$\frac{C_1 z}{z - p_1} + \frac{C_2 z}{z - p_2} = \frac{(C_1 + C_2)z^2 - (C_1 p_2 + C_2 p_1)z}{z^2 - (p_1 + p_2)z + p_1 p_2} \tag{4.66}$$

$$= \frac{C_1 + C_2 - (C_1 p_2 + C_2 p_1)z^{-1}}{1 - (p_1 + p_2)z^{-1} + p_1 p_2 z^{-2}} \tag{4.67}$$

比较4.65b式和4.67式，在4.65b式中令 $i = 1$，我们求得

$$\begin{aligned} a_{01} &= C_1 + C_2, & a_{11} &= -(C_1 p_2 + C_2 p_1) \\ b_{11} &= -(p_1 + p_2), & b_{21} &= p_1 p_2 \end{aligned} \tag{4.68}$$

如果我们利用事实 $p_2 = p_1^*$ 和 $C_2 = C_1^*$，代入 p_1 和 C_1 的值，那么

$$a_{01} = C_1 + C_1^* = 2 \times 0.114\,383 = 0.2288$$

$$\begin{aligned} a_{11} &= -(C_1 p_1^* + C_1^* p_1) \\ &= -(|C_1|\text{e}^{\text{j}\theta_1}|p_1|\text{e}^{-\text{j}\phi_1} + |C_1|\text{e}^{-\text{j}\theta_1}|p_1|\text{e}^{\text{j}\phi_1}) \\ &= -|C_1||p_1|[\text{e}^{\text{j}(\theta_1 - \phi_1)} + \text{e}^{-\text{j}(\theta_1 - \phi_1)}] \\ &= -2|C_1||p_1|\cos(\theta_1 - \phi_1) \\ &= -2 \times 0.676\,410\,4 \times 1 \cos(1.400\,877 - 0.785\,400\,68) \\ &= -1.1046 \end{aligned} \tag{4.69}$$

（其中 $\theta_1 = \angle C_1$，$\phi_1 = \angle p_1$）。于是我们有

$$H_1(z) = \frac{0.2288 - 1.1046z^{-1}}{1 - 1.4142z^{-1} + z^{-2}} \tag{4.70}$$

其中分母系数的值直接取自 4.63 式。类似地，由含有 C_3 和 C_4 的部分分式，我们有

$$H_2(z) = \frac{-1.2288 - 0.0335z^{-1}}{1 + z^{-1} + 0.5z^{-2}} \tag{4.71}$$

合并结果得

$$H(z) = 2 + \frac{0.2288 - 1.1046z^{-1}}{1 - 1.4142z^{-1} + z^{-2}} + \frac{-1.2288 - 0.0335z^{-1}}{1 + z^{-1} + 0.5z^{-2}}$$

尽管上面的过程很简单，但是表达式非常繁琐，且容易出现错误，特别是如果部分分式系数是手工计算的。我们在附录4B描述了可以用来在给定串行形式的传递函数时求取并行结构系数的C语言程序。这一程序实际上是在同一附录中描述的部分分式展开程序的简单扩展。在第 8 章中，我们将详细讨论串行和并行实现结构的应用。

4.6 小结

z变换的知识在DSP中是非常重要的，对于表示、分析和设计离散系统来说，它是一个价值无法衡量的工具。

我们已经说明了如何计算离散时间序列的z变换，以及如何从z变换恢复序列。本章提供了几个C语言和MATLAB程序，可以使读者加深对概念以及z变换在信号处理中的应用的实际理解。请尽可能地应用这些程序。

习题

4.1　求下列离散时间序列的 z 变换：

(1)　$x(n) = \sin(n\omega T),\ n = 0, 1, \ldots$

(2)　$x(n) = a^n,\quad n \geqslant 0$

$\quad\quad = 0,\quad n < 0$

(3)　$x(n) = 1,\quad 0 \leqslant n \leqslant N-1$

$\quad\quad = 0,$　其他

4.2　一个指数序列定义为

$$x(n) = \mathrm{e}^{-kn},\quad n \geqslant 0$$

求它的 z 变换，包括在下列情况下 z 变换收敛的约束条件：

(1) k 是实数；

(2) k 是复数。

4.3　给定一个因果序列 $x(n)$和 $nx(n)$，它们的 z 变换分别为 $X(z)$和 $X'(z)$，证明：

$$X'(z) = -z\frac{\mathrm{d}X(z)}{\mathrm{d}z}$$

4.4　离散时间序列的 z 变换为

$$X(z) = \sum_{n=0}^{\infty} x(n)z^{-n}$$

从上式开始证明 z 反变换为

$$x(n)=\frac{1}{2\pi j}\oint z^{n-1}X(z)\,dz,\quad n>0$$

阐述所做的任何假定，简要讨论留数定理在计算以上积分中的作用。

4.5 (1) 用级数展开的方法求对应于下列 z 变换的因果离散时间序列的前五个值：

(a) $X(z)=\dfrac{z-1}{(z-0.7071)^2}$

(b) $X(z)=\dfrac{1}{(z-0.5)(z+0.9)^3}$

(c) $X(z)=\dfrac{z^4-1}{z^4+1}$

(d) $X(z)=\dfrac{z^3-z^2+z-1}{(z+0.9)^3}$

(2) 用部分分式展开法重新做部分(1)。

(3) 用留数法重新做部分(1)。

4.6 (1) 给定下列形式的 z 变换：

$$X(z)=\frac{N(z)}{D(z)}$$

其中 $N(z)$ 和 $D(z)$ 是多项式，假定 $X(z)$ 在 $z=p_k$ 处有一个极点，证明：

$$\mathrm{Res}[X(z),p_k]=\frac{N(p_k)}{D'(p_k)}$$

其中

$$D'(p_k)=\frac{dD(z)}{dz}$$

(2) 利用这一结果求下面 z 变换的反变换，

$$X(z)=\frac{1}{z^4-1}$$

4.7 对于一个稳定的因果系统，给定下列传递函数，用留数法求闭合形式的冲激响应 $h(n)$：

$$H(z)=\frac{1}{1+b_1z^{-1}+b_2z^{-2}}$$

假定所有极点都不同，且都为复数。

4.8 N 阶离散时间系统的 z 变换的部分分式展开为

$$X(z)=\frac{N(z)}{D(z)}=B_0+\sum_{k=1}^{N}\frac{C_kz}{z-p_kz^k}$$

其中

$$N(z)=a_0+a_1z^{-1}+a_2z^{-2}+\ldots+a_Nz^{-N}$$
$$D(z)=b_0+b_1z^{-1}+b_2z^{-2}+\ldots+b_Mz^{-M}$$

p_k 是 $X(z)$ 的极点（假定不同），C_k 是部分分式的系数。根据极点 p_k 和 $N(z)$，求 C_k（$k=1,2,\ldots$）的一般表达式。假定 $N=3$，用长除法证明 B_0 为

$$B_0=a_3/b_3$$

4.9　给定差分方程：

$$y(n) + B_1 y(n-1) + B_2 y(n-2) = A, \quad n \geqslant 0$$

其中 A、B_1 和 B_2 是任意常数，求 z 变换 $Y(z)$ 的表达式。应用合适的 z 反变换技术求 $y(n)$ 的闭合表达式。

4.10　一个二阶离散时间系统由下列 z 变换函数刻画，

$$H(z) = \frac{1}{(z-0.9)^2}, \quad |z| > 0.9$$

用留数法求对应的离散时间序列 $h(n)$。

4.11　对于图 4.17 所示的离散时间系统，求联系输出 $y(n)$ 和输入 $x(n)$ 的差分方程，推导它的传递函数 $H(z)$。

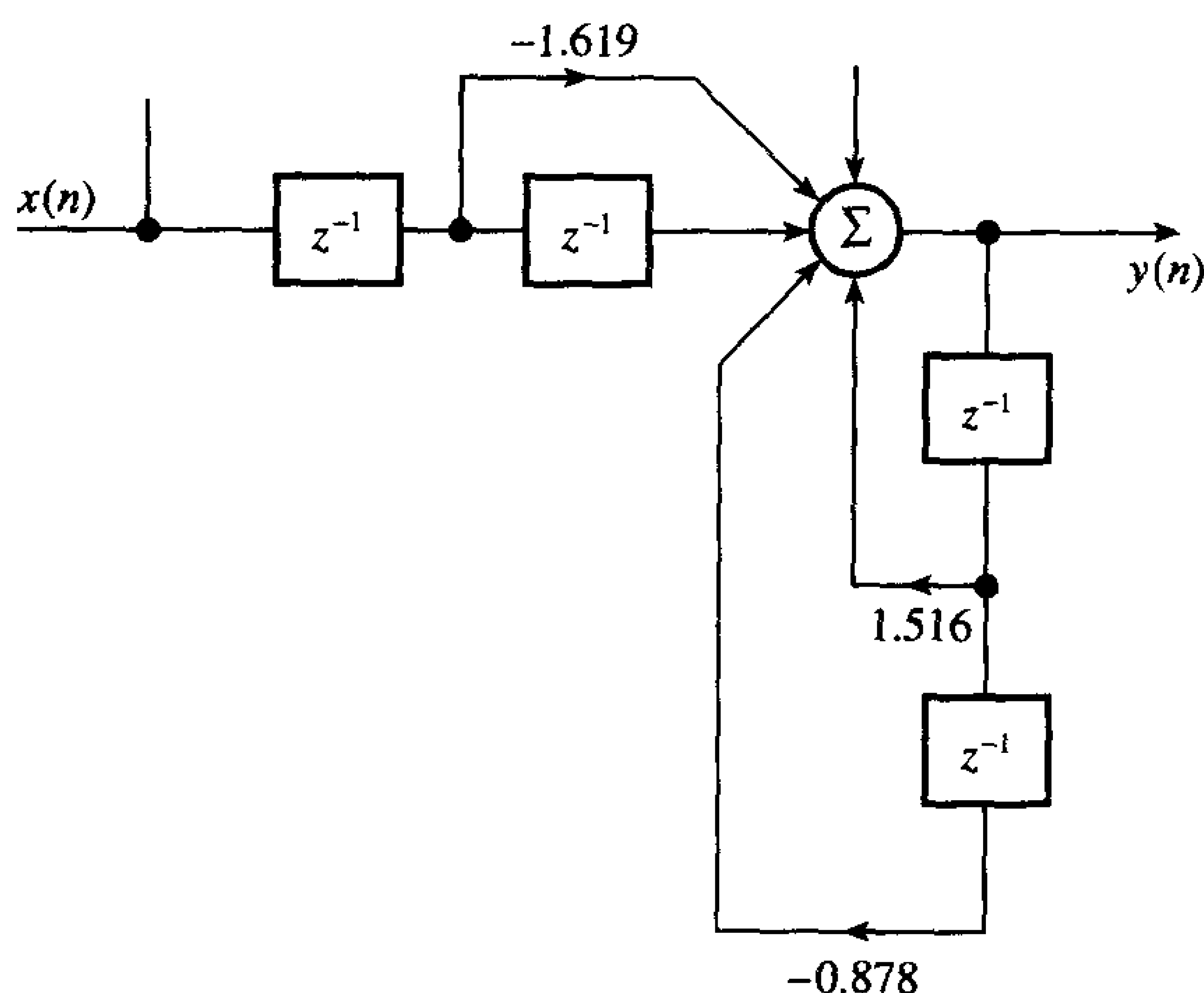


图 4.17　习题 4.11 的离散时间系统的框图

4.12　离散时间系统的传递函数在 $z = 0.5$、$z = 0.1 \pm \mathrm{j}0.2$ 处有两个极点以及在 $z = -1$、$z = 1$ 处有两个零点。

(1) 画出系统的极零图。

(2) 从极零图推导系统的传递函数 $H(z)$。

(3) 建立系统的差分方程。

(4) 用信号流图形式画出系统的实现图。

4.13　离散时间系统的信号流图如图 4.18 所示。求联系输出 $y(n)$ 与输入 $x(n)$ 的二步差分方程，由差分方程推导传递函数 $H(z)$。

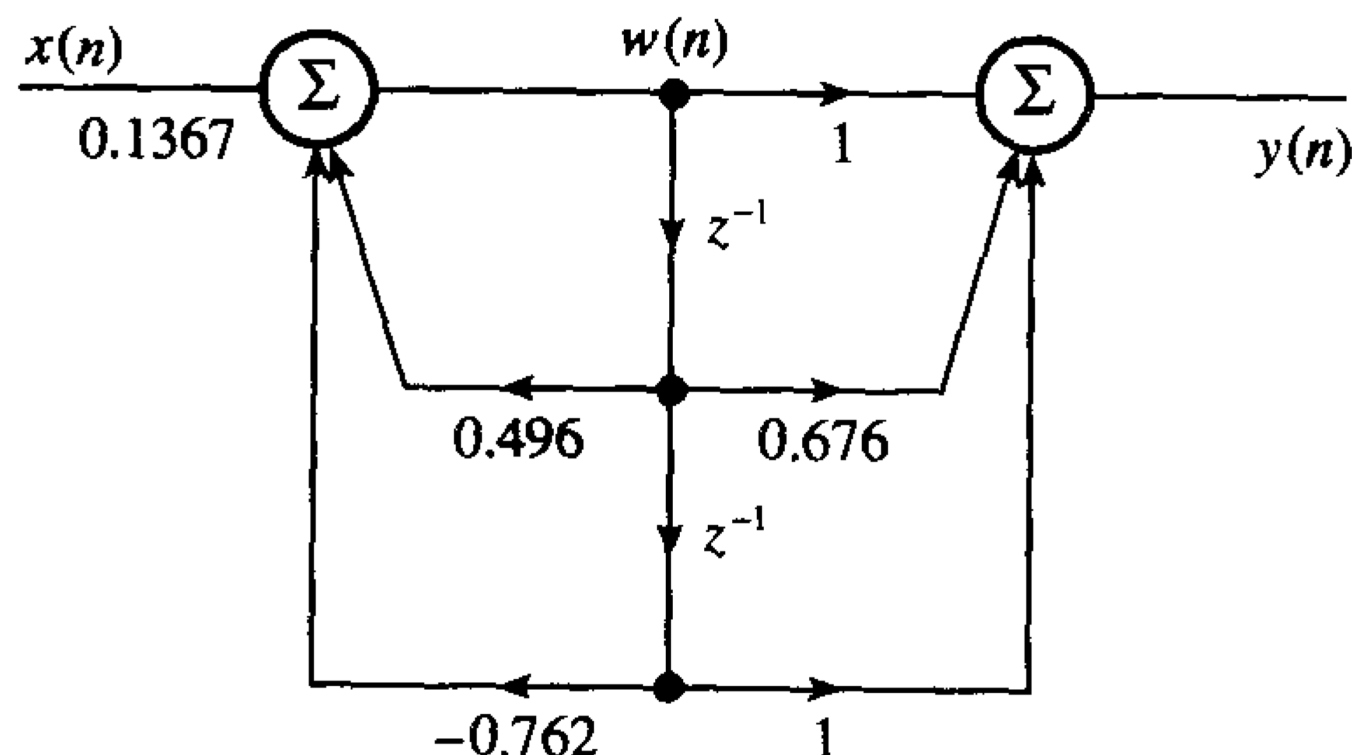


图 4.18　习题 4.13 的离散时间系统的信号流图

4.14 对于一个归一化形式的带通离散时间滤波器的频率响应指标如下：

通带	$0.4\pi \sim 0.6\pi$
阻带	$0 \sim 0.3\pi$ 和 $0.7\pi \sim \pi$
抽样间隔	$T = 100\ \mu s$

(1) 用弧度表示技术指标（非归一化）。

(2) 将技术指标从弧度转换成标准的 Hz 单位。

(3) 将技术指标从(2) 的 Hz 单位转换成归一化形式。

(4) 对于上面三种情况，画出从 0 到抽样频率范围内的频率响应。

4.15 一个由下列 z 变换刻画的 LTI 系统：

$$\frac{1 + z^{-2}}{1 + 0.81z^{-2}}$$

求在 dc、1/4 和 1/2 抽样频率处的频率响应，画出在区间 $0 \leqslant \omega \leqslant \omega_s$ 上的频率响应，其中 ω_s 是抽样频率，单位为弧度 / 秒。

4.16 对于一个具有如下技术指标的简单的低通离散滤波器，需要满足一定的要求，

截止频率	1 kHz
抽样频率	10 kHz

画出滤波器合适的极零图。

根据极零图求滤波器的传递函数。求在 1 kHz、2.5 kHz 和 5 kHz 处的幅度响应和相位响应，画出幅度频率响应。

4.17 某系统的传递函数为

$$H(z) = \frac{(1 - 1.094\,621z^{-1} - z^{2})(1 - 0.350\,754z^{-1} + z^{-2})}{(1 - 1.340\,228z^{-1} + 0.796\,831z^{-2})(1 - 0.5z^{-1} - 0.5z^{-2})}$$

(1) 求极点和零点，画出极零图。

(2) 说明系统是否稳定。

应用 MATLAB 和语言工具的基于计算机的习题

4.18 离散时间系统的传递函数为

$$H(z) = \frac{z^2 - z}{z^2 - 0.9051z + 0.4096}$$

(a) 借助 MATLAB 的 roots 命令求极点和零点的位置（零点应该在 $z = 0$、$z = \pm 1$，极点在 $0.64 \angle 45°$）。

(b) 如果分子与分母多项式按 z 的负幂增加的形式表示，系统的传递函数表示为

$$H(z) = \frac{1 - z^{-1}}{1 - 0.9051z^{-1} + 0.4096z^{-2}}$$

重复(a)。

(c) 比较(a)和(b)的结果，对任何差别做出解释，如何才能保证两者的答案相同。

(d) 利用 MATLAB 的 zplane 命令画出(a)中的 $H(z)$在下面两种情况的极零图：

(i) 用分子和分母 $b(z)$和 $a(z)$的系数作为输入。

(ii) 用 $H(z)$的极点和零点的位置作为输入。

4.19 离散时间陷波滤波器的传递函数为

$$H(z)=\frac{1-2\cos\theta+z^{-2}}{1-2r\cos\theta z^{-1}+r^2z^{-2}}$$

其中 r 是极点的半径，θ 是极点和零点的角度。

(a) 对下列几种情况，用 MATLAB 画出滤波器的幅度频率特性，估计凹口深度（相对于 0 Hz 处的幅度响应）：

(i) $r=0.5,\ \theta=\pm15^\circ$

(ii) $r=0.5,\ \theta=\pm60^\circ$

(iii) $r=0.5,\ \theta=\pm90^\circ$

(iv) $r=0.5,\ \theta=\pm120^\circ$

(b) 对下列几种情况，用 MATLAB 画出滤波器的幅度频率特性，估计凹口深度（相对于 0 Hz 处的幅度响应）：

(v) $r=0.5,\ \theta=\pm45^\circ$

(vi) $r=0.8,\ \theta=\pm45^\circ$

(vii) $r=0.9,\ \theta=\pm45^\circ$

(viii) $r=0.99,\ \theta=\pm45^\circ$

解释极点和零点的位置（半径和角度）是如何影响凹口滤波器的频率响应的。

4.20 一个离散时间系统的 z 变换为

$$H(z)=\sum_{k=0}^{8}a_kz^{-k}\Big/\sum_{k=0}^{8}b_kz^{-k}$$

其中

$$\begin{aligned}
&a_0=2.740\,584\times10^{-2} && b_0=1\\
&a_1=2.825\,341\times10^{-3} && b_1=2.233\,030\times10^{-1}\\
&a_2=-2.932\,353\times10^{-2} && b_2=2.353\,762\\
&a_3=3.563\,199\times10^{-4} && b_3=4.369\,285\times10^{-1}\\
&a_4=4.924\,136\times10^{-2} && b_4=2.712\,411\\
&a_5=3.563\,226\times10^{-4} && b_5=3.571\,619\times10^{-1}\\
&a_6=-2.932\,353\times10^{-2} && b_6=1.593\,957\\
&a_7=2.825\,337\times10^{-3} && b_7=1.141\,820\times10^{-1}\\
&a_8=2.740\,582\times10^{-2} && b_8=4.143\,201\times10^{-1}
\end{aligned}$$

借助附录中描述的 z 反变换程序或者合适的 MATLAB 程序（幂级数的方法），求出并且画出系统的冲激响应。根据你画出的图阐述系统是稳定的、临界稳定的或者不稳定的？

4.21 用因式分解形式表示的三阶 IIR 系统的传递函数为

$$H(z)=\frac{N_1(z)N_2(z)}{D_1(z)D_2(z)}$$

其中

$$\begin{aligned}
&N_1(z)=1-0.971\,426z^{-1}+z^{-2}\\
&N_2(z)=1+z^{-1}\\
&D_1(z)=1-0.935\,751z^{-1}+0.726\,879z^{-2}\\
&D_2(z)=1+0.183\,11z^{-1}
\end{aligned}$$

(1) 对于一个用二阶和一阶项表示的串联结构，画出系统的实现图。

(2) 将 $H(z)$表示成部分分式和的形式：

$$H(z) = B_0 + \sum_{k=1}^{3} \frac{C_k}{z - p_k}$$

并且利用附录4C的部分分式展开程序或者合适的MATLAB程序来计算系数B_0和C_k。

(3) 组合部分分式，使系统能够通过一阶和一个二阶项利用并行结构实现。

(4) 应用从(3) 得出的结果，画出系统并行实现的结构图。

4.22 一个由下列 z 变换刻画的数字凹口滤波器：

$$\frac{z^2+1}{z^2+r^2}$$

对于下面每种情况，用 FFT 方法和 1 kHz 的抽样频率，估计频率响应。(i) $r = 0.8$; (ii) $r = 0.95$; (iii) $r = 1$。解释你的结果。

4.23 一个低通的离散时间滤波器具有下列传递函数：

$$H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_4 z^{-4}}{b_0 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_4 z^{-4}}$$

其中

$a_0 = 0.193\,441$　$b_0 = 1$

$a_1 = 0.378\,331$　$b_1 = -2.516\,884$

$a_2 = 0.524\,14$　$b_2 = 1.054\,118$

$a_3 = 0.378\,331$　$b_3 = -0.240\,603$

$a_4 = 0.193\,441$　$b_4 = 0.198\,586\,1$

利用下面的方法估计滤波器的频率响应：

(1) 附录 4C 中讨论的直接频率响应的计算程序；

(2) 幂级数法和 FFT 法。

比较两个结果，并解释任何差别。

4.24 简单带通 FIR 系统的系数如表 4.4 所示。假定抽样频率为 10 kHz，用附录 4C 讨论的程序来计算系统的幅度频率响应。

表 4.4　习题 4.24 的 FIR 低通滤波器的系数

```
H (1) = -0.67299600E-02 = H(35)
H (2) =  0.16799420E-01 = H(34)
H (3) =  0.17195700E-01 = H(33)
H (4) = -0.27849080E-01 = H(32)
H (5) = -0.17486810E-01 = H(31)
H (6) =  0.13515580E-01 = H(30)
H (7) =  0.45570510E-02 = H(29)
H (8) =  0.33293060E-01 = H(28)
H (9) =  0.95162150E-02 = H(27)
H(10) = -0.68548560E-01 = H(26)
H(11) = -0.68992230E-02 = H(25)
H(12) =  0.23802370E-01 = H(24)
H(13) = -0.11597510E-01 = H(23)
H(14) =  0.12073780E+00 = H(22)
H(15) =  0.23806900E-01 = H(21)
H(16) = -0.29095690E+00 = H(20)
H(17) = -0.12362380E-01 = H(19)
H(18) =  0.36717700E+00 = H(18)
```

4.25 给定 z 变换：

$$H(z)=\frac{1+3z^{-1}+z^{-2}+z^{-3}}{1+(1-k)z^{-1}+(k+0.3561)z^{-2}+0.3561k}$$

利用附录4B中的幂级数展开法的计算机程序，对下列几种情况计算足够数量的系统冲激响应的值：

(1) $k=-1$;

(2) $k=1$;

(3) $k=2$;

(4) $k=0.9$。

对于每种情况，画出冲激响应，系统是稳定的、临界稳定的还是不稳定的？

4.26 编写一个检查部分分式展开结果的C语言程序。程序应该接受极点 p_k（$k=1,2,\ldots,M$）值和有关的系数 C_k（$k=1,2,\ldots,M$）作为输入，得到 $A(z)$ 和 $B(z)$ 的系数作为输出，其中

$$X(z)=\frac{A(z)}{B(z)}$$

和

$$A(z)=a_0+a_1z^{-1}+a_2z^{-2}+\ldots+a_Nz^{-N}$$

$$B(z)=b_0+b_1z^{-1}+b_2z^{-2}+\ldots+b_Mz^{-M}$$

把程序扩展到检查将串联结构转换成并联结构的结果。

4.27 使用 MATLAB 重新做习题 4.26。

参考文献

Atkinson L.V. and Harley P.J. (1983) *An Introduction to Numerical Methods with Pascal*, Chapter 3. Wokingham: Addison-Wesley.

Jury E.I. (1964) *Theory and Applications of the z-transform Method.* New York: Wiley.

Mathews J.H. (1982) *Basic Complex Variables for Mathematics and Engineering.* Boston MA: Allyn and Bacon.

Proakis J.G. and Manolakis D.G. (1992) *Digital Signal Processing*, 2nd edn. New York: Macmillan.

参考书目

Ahmed N. and Natarajan T. (1983) *Discrete-time Signals and Systems.* Reston VA: Reston Publishing Co. Inc.

Churchhill R.V., Brown J.W. and Verhey R.F. (1976) *Complex Variables and Applications.* New York: McGraw-Hill.

Jong M.T. (1982) *Methods of Discrete Signals and Systems Analysis.* New York: McGraw-Hill.

Oppenheim A.V. and Schafer R.W. (1975) *Digital Signal Processing.* Englewood Cliffs NJ: Prentice-Hall.

Rabiner L.R. and Gold B. (1975) *Theory and Application of Digital Signal Processing.* Englewood Cliffs NJ: Prentice-Hall.

Ragazzini J.R. and Zadeh L.A. (1952) Analysis of sampled data systems. *Trans. AIEE*, **71**(II), 225–34.

Steiglitz K. (1974) *An Introduction to Discrete Systems.* New York: Wiley.

Strum R.D. and Kirk D.E. (1988) *First Principles of Discrete Systems and Digital Signal Processing.* Reading MA: Addison-Wesley.

附录

4A z反变换的递归算法

在本章已经提到长除法可以用递归形式重写。特别是我们要在这里证明，对于给定的z变换$X(z)$，即

$$X(z)=\frac{b_0+b_1z^{-1}+b_2z^{-2}}{a_0+a_1z^{-1}+a_2z^{-2}}$$

z反变换$x(n)$可以求得为（Jury, 1964）

$$x(n)=\frac{1}{a_0}\left[b_n-\sum_{i=1}^{n}x(n-i)a_i\right],\quad n=1,2,\ldots$$

$$x(0)=\frac{b_0}{a_0}$$

结果可以推广。利用长除法，我们可以将$X(z)$表示为幂级数的形式：

$$\frac{b_0}{a_0}+\left[\left(b_1-\frac{b_0}{a_0}a_1\right)\Big/a_0\right]z^{-1}+\frac{1}{a_0}\left[\left(b_2-\frac{b_0}{a_0}a_2\right)-\frac{a_1}{a_0}\left(b_1-\frac{b_0}{a_0}a_1\right)\right]z^{-2}$$

$$a_0+a_1z^{-1}+a_2z^{-2}\;\overline{\big)\;b_0+b_1z^{-1}+b_2z^{-2}}$$

$$b_0+\left(\frac{b_0}{a_0}a_1\right)z^{-1}\qquad+\left(\frac{b_0}{a_0}a_2\right)z^{-2}$$

$$\overline{\left(b_1-\frac{b_0}{a_0}a_1\right)z^{-1}+\left(b_2-\frac{b_0}{a_0}a_2\right)z^{-2}}$$

$$\left(b_1-\frac{b_0}{a_0}a_1\right)z^{-1}+\frac{a_1}{a_0}\left(b_1-\frac{b_0}{a_0}a_1\right)z^{-2}+\frac{a_2}{a_0}\left(b_1-\frac{b_0}{a_0}a_1\right)z^{-3}$$

$$\overline{\left[\left(b_2-\frac{b_0}{a_0}a_2\right)-\frac{a_1}{a_0}\left(b_1-\frac{b_0}{a_0}a_1\right)\right]z^{-2}-\frac{a_2}{a_0}\left(b_1-\frac{b_0}{a_0}a_1\right)z^{-3}}$$

$$\left[\left(b_2-\frac{b_0}{a_0}a_2\right)-\frac{a_1}{a_0}\left(b_1-\frac{b_0}{a_0}a_1\right)\right]z^{-2}+\frac{a_1}{a_0}\left[\left(b_2-\frac{b_0}{a_0}a_2\right)\right.$$

$$\left.-\frac{a_1}{a_0}\left(b_1-\frac{b_0}{a_0}a_1\right)\right]z^{-3}+\frac{a_2}{a_0}\left[\left(b_2-\frac{b_0}{a_0}a_2\right)-\frac{a_1}{a_0}\left(b_1-\frac{b_0}{a_0}a_1\right)\right]z^{-4}$$

$$\overline{\left\{\left[-\frac{a_2}{a_0}\left(b_1-\frac{b_0}{a_0}a_1\right)\right]-\frac{a_1}{a_0}\left[\left(b_2-\frac{b_0}{a_0}a_2\right)\right.\right.}$$

$$\left.\left.-\frac{a_1}{a_0}\left(b_1-\frac{b_0}{a_0}a_1\right)\right]\right\}z^{-3}-\frac{a_2}{a_0}\left[\left(b_2-\frac{b_0}{a_0}a_2\right)\right.$$

$$\left.-\frac{a_1}{a_0}\left(b_1-\frac{b_0}{a_0}a_1\right)\right]z^{-4}$$

$$\vdots$$

长除法的商给出了幂级数的系数：

$$X(z) = \frac{b_0}{a_0} + \left[\left(b_1 - \frac{b_0}{a_0}a_1\right)\Big/a_0\right]z^{-1} + \frac{1}{a_0}\left[\left(b_2 - \frac{b_0}{a_0}a_2\right) - \frac{a_1}{a_0}\left(b_1 - \frac{b_0}{a_0}a_1\right)\right]z^{-2}$$

$$+ \frac{1}{a_0}\left\{\left[-\frac{a_2}{a_0}\left(b_1 - \frac{b_0}{a_0}a_1\right)\right] - \frac{a_1}{a_0}\left[\left(b_2 - \frac{b_0}{a_0}a_2\right) - \frac{a_1}{a_0}\left(b_1 - \frac{b_0}{a_0}a_1\right)\right]\right\}z^{-3} + \ldots$$

由因果系统 z 变换的定义，$X(z)$为

$$X(z) = \sum_{n=0}^{\infty} x(n)z^{-n} = x(0) + x(1)z^{-1} + x(2)z^{-2} + \ldots$$

因此，我们可以写成

$$x(0) = \frac{b_0}{a_0}$$

$$x(1) = \frac{1}{a_0}\left(b_1 - \frac{b_0}{a_0}a_1\right) = \frac{1}{a_0}[b_1 - x(0)a_1]$$

$$x(2) = \frac{1}{a_0}\left[\left(b_2 - \frac{b_0}{a_0}a_2\right) - \frac{a_1}{a_0}\left(b_1 - \frac{b_0}{a_0}a_1\right)\right]$$

$$= \frac{1}{a_0}[b_2 - x(0)a_2 - a_1x(1)]$$

$$x(3) = \frac{1}{a_0}\left[\left\{-\frac{a_2}{a_0}\left(b_1 - \frac{b_0}{a_0}a_1\right)\right\} - \frac{a_1}{a_0}\left\{\left(b_2 - \frac{b_0}{a_0}a_2\right) - \frac{a_1}{a_0}\left(b_1 - \frac{b_0}{a_0}a_1\right)\right\}\right]$$

$$= \frac{1}{a_0}[-a_2x(1) - a_1x(2)]$$

总之，我们可以写成

$$x(n) = \frac{1}{a_0}\left[b_n - \sum_{i=1}^{n} x(n-i)a_i\right], \quad n = 1, 2, \ldots$$

$$x(0) = b_0/a_0$$

4B　计算 z 反变换以及串行到并形结构转换的 C 程序

我们已经建立了用幂级数和部分分式展开法来计算z反变换的C语言程序，这个程序也可以用来将离散时间系统传递函数$H(z)$从串行结构转化为并行结构。这个程序很大，所以为了方便将其组成两个程序模块 izt.c 和 ltilib.c，保存在两个分开的文件中，并分别进行编译，然后进行链接：

izt.c　通过幂级数或部分分式展开计算z反变换中，以及通过部分分式展开将串行形式的传递函数 $H(z)$转换成等效的并行形式。

ltilib.c　包含 power_series 和 partial_fraction 函数的函数库。

程序和库由于缺乏空间在这里没有列出清单，但是可以在指导手册 *A Practical Guide for MATLAB and C Language Implementations of DSP Algorithms*（详细内容请参见前言）的CD上找到。

4B.1 幂级数法

根据下列方程，由函数 power_series()（参见程序）递推计算 z 反变换 $x(n)$：

$$x(n)=\left[b_n-\sum_{i=1}^{n}x(n-i)a_i\right]\Big/a_0,\quad n=1,2,\ldots \tag{4B.1a}$$

其中

$$x(0)=b_0/a_0 \tag{4B.1b}$$

为了利用程序通过幂级数法计算 z 反变换，z 变换可以用直接的形式或者串联形式表示：

$$X(z)=\frac{b_0+b_1z^{-1}+b_2z^{-2}+\ldots+b_Nz^{-N}}{a_0+a_1z^{-1}+a_2z^{-2}+\ldots+a_Mz^{-M}}\quad \text{直接形式} \tag{4B.2a}$$

$$X(z)=\prod_{k=i}^{K}X_i(z)\quad \text{串联形式} \tag{4B.2b}$$

其中 $X_i(z)$是由下式给出的二阶项：

$$X_i(z)=\frac{b_{0i}+b_{1i}z^{-1}+b_{2i}z^{-2}}{1+a_{1i}z^{-1}+a_{2i}z^{-2}} \tag{4B.3}$$

必须产生名为coeff.dat的输入数据文件，文件包含级数K（对于直接形式，K=1）、z变换分母和分子的系数。输入数据文件的使用是很方便的，因为它消除了键入系数时可能造成的错误。此外，它也使我们更易于将数据输入到其他程序。下面的例子说明了应用程序通过幂级数法计算z反变换。

例 4B.1 用幂级数法求由下列 z 变换所刻画的离散时间系统的 z 反变换的前五个值：

$$X(z)=\frac{0.183\,301\,5+0.341\,956\,1z^{-1}+0.341\,956\,1z^{-2}+0.183\,301\,5z^{-3}}{1-0.352\,518\,2z^{-1}+0.419\,402\,3z^{-2}-0.016\,369z^{-3}}$$

很清楚，$X(z)$是用直接形式表示的，输入数据文件用所有PC机都带的 edlin 建立，它的形式如下：

```
1
1  -0.3525182  0.4194023  -0.016369
0.1833015  0.3419561  0.3419561  0.1833015
```

程序的输出总结如下：

$h(0)=-0.016\,369$; $h(1)=0.177\,531$; $h(2)=0.411\,404$; $h(3)=0.070\,570\,5$;

$h(4)=-0.147\,666\,6$

例 4B.2 用幂级数法求由下列系统的 z 反变换的前五个值：

$$X(z)=\frac{N_1(z)N_2(z)N_3(z)}{D_1(z)D_2(z)D_3(z)}$$

其中

$$N_1(z)=1-1.122\,346z^{-1}+z^{-2}$$

$$N_2(z)=1-0.437\,833z^{-1}+z^{-2}$$

$$N_3(z)=1+z^{-1}$$

$$D_1(z)=1-1.433\,509z^{-1}+0.858\,110z^{-2}$$

$$D_2(z)=1-1.293\,601z^{-1}+0.556\,929z^{-2}$$

$$D_3(z)=1-0.612\,159z^{-1}$$

很显然，传递函数由三级组成：两级为二阶系统，一级为一阶系统。一阶系统当做二阶系统在 z^{-2} 项的系数为零来输入，输入数据文件如下：

```
3                               /* 级数; 最大为5*/
1  -1.433509    0.858110        /*D1(z)的系数 */
1  -1.122346    1               /*N1(z)的系数 */
1  -1.293601    0.556929        /*D2(z)的系数 */
1  -0.437833    1               /*D2(z)的系数 */
1  -0.6121593   0               /*D3(z)的系数 */
1  1            0               /*N3(z)的系数 */
```

右边的注释不是文件的一部分，它只是用于说明，程序的输出总结如下：

$$x(0)=1;\ x(1)=2.779\,09;\ x(2)=5.2725$$

$$x(3)=8.7218;\ x(4)=11.7438;\ x(5)=13.4723$$

4B.2　部分分式展开

给定一个具有不同极点的 N 阶 z 变换，即

$$X(z)=\frac{N(z)}{D(z)}=\frac{b_0z^N+b_1z^{N-1}+\ldots+b_{N-1}z+b_N}{a_0z^N+a_1z^{N-1}+\ldots+a_{N-1}z+a_N}$$

那么 $X(z)$ 可以展开成部分分式的形式：

$$\frac{N(z)}{zD(z)}=\frac{N(z)}{z(z-p_1)(z-p_2)(z-p_3)\ldots(z-p_N)}=\frac{B_0}{z}+\sum_{k=1}^{M}\frac{C_k}{z-p_k}\tag{4B.4}$$

其中

$$N(z)=b_0z^N+b_1z^{N-1}+\ldots+b_{N-1}z+b_N$$

$$D(z)=a_0z^N+a_1z^{N-1}+\ldots+a_{N-1}z+a_N$$

p_k 是 $X(z)$ 的极点（假定一阶），C_k 是部分分式的系数，常数 B_0 为

$$B_0=b_N/a_N\tag{4B.5}$$

与 p_k 有关的部分分式系数 C_k 可以通过对 4B.4 式的两边乘以 $z-p_k$ 并令 $z=p_k$ 得到：

$$C_k=\frac{N(z)(z-p_k)}{zD(z)}=\left.\frac{N(z)}{zD_k(z)}\right|_{z=p_k}\tag{4B.6}$$

其中

$$D_k(z)=\prod_{\substack{i=1\\i\neq k}}^{M}(z-p_i)$$

例如，为了求 C_1，我们在 4B.4 式的两边乘以 $z-p_1$ 并令 $z=p_1$ 得到：

$$C_1=\frac{N(z)(z-p_1)}{zD(z)}=\left.\frac{N(z)\cancel{(z-p_1)}}{z\cancel{(z-p_1)}(z-p_2)(z-p_3)\ldots(z-p_N)}\right|_{z=p_1}=\left.\frac{N(z)}{zD_1(z)}\right|_{z=p_1}$$

其中

$$D_1(z)=(z-p_2)(z-p_3)\ldots(z-p_N)$$

利用极点的极坐标表达式，即 $p_k=r_k\mathrm{e}^{\mathrm{j}\theta k}$，系数可以表示为

$$C_k=\frac{N(r_k\mathrm{e}^{\mathrm{j}\theta k})}{r_k\mathrm{e}^{\mathrm{j}\theta k}D_k(\mathrm{e}^{\mathrm{j}\theta k})}\quad k=1,\ldots,N\tag{4B.7}$$

部分分式展开函数首先求极点 p_k（$k=1,2,\ldots,N$）的位置，然后对每个极点计算 4B.7 式。

当B_0和C_k求得以后，z变换可以写成为

$$X(z)=B_0+\sum_{k=1}^{N}\frac{C_1 z}{z-p_k} \tag{4B.8}$$

对于因果序列，z反变换是4B.8式中每一项的z反变换之和：

$$x(n)=B_0u(n)+C_1(p_1)^n+C_2(p_2)^n+\ldots+C_N(p_N)^n \tag{4B.9}$$

为了利用程序来计算部分分式系数，z变换必须应用二阶因子用串联形式表示，通过一个例子就可以清楚地看到这一点。

例 4B.3 应用部分分式展开法求4B.2式给出的五阶传递函数的z反变换。

传递函数的部分分式展开具有如下形式：

$$X(z)=B_0+\sum_{k=1}^{5}\frac{C_k z}{z-p_k} \tag{4B.10}$$

本例的输入数据文件coeff.data与例4B.2的相同，程序的输出总结如下：

z变换的极点

pk	real	imag	mag	phase
1	0.716754	0.586833	0.926342	39.308436
2	0.716754	−0.586833	0.926342	−39.308436
3	0.646801	0.372261	0.746277	29.922232
4	0.646801	−0.372261	0.746277	−29.922232
5	0.612159	0.000000	0.612159	0.000000

部分分式系数

B0 = −3.418163

Ck	real	imag	mag	phase
1	1.611473	5.209672	5.453212	72.811944
2	1.611473	−5.209672	5.453212	−72.811943
3	−19.580860	−9.681908	21.843751	−153.689550
4	−19.580861	9.681908	21.843751	153.689551
5	40.356939	0.000000	40.356939	0.000000

由这些值得到z反变换为

$$x(n)=B_0u(n)+\sum_{k=1}^{5}C_k(p_k)^n,\quad n\geqslant 0$$

对复共轭极点进行组合，用表4.1也可以求z反变换$x(n)$。这一问题留给读者练习。

4B.3 串联到并联结构的转换

根据例4.14描述的原则，程序也可以用来将z变换从串联结构转换成并联形式。

例 4B.4 用串联形式给出的四阶离散时间系统的传递函数为

$$H(z)=\frac{N_1(z)N_2(z)}{D_1(z)D_2(z)}$$

其中

$$D_1(z)=1+0.052\,921z^{-1}+0.831\,73z^{-2}$$

$$N_1(z)=1+0.481\,199z^{-1}+z^{-2}$$

$$D_2(z)=1-0.304\,609z^{-1}+0.238\,865z^{-2}$$

$$N_2(z)=1+1.474\,597z^{-1}+z^{-2}$$

用程序将传递函数从串联结构转换成并联形式。

输入数据文件具有下列形式：

```
2
1  0.05292   0.83173
1  0.481199   1
1  -0.304609   0.238865
1  1.474597   1
```

由程序给出了下列输出：

```
selected desired operation
0          for power series method of IZT
1          for partial fraction coeffs estimation
2          for cascade to parallel conversion
2
poles of the z-transform
pk         real          imag          mag          phase
1          -0.026460     0.911413      0.911797     91.662967
2          -0.026460     -0.911413     0.911797     -91.662967
3          0.152305      0.464401      0.488738     71.842631
4          0.152305      -0.464401     0.488738     -71.842631
partial fraction coeffs
B0=5.035604
Ck         real          imag          mag          phase
1          -0.257338     0.421333      0.493705     121.415410
2          -0.257338     -0.421333     0.493705     -121.415409
3          -1.760464     -3.766287     4.157421     -115.052650
4          -1.760464     3.766287      4.157421     115.052650
press enter to continue
stage      Ni(z)
0          -0.514677     -0.781635
1          -3.520927     4.034388
2          0.000000      -0.000000
stage      Di(z)
0          1.000000      0.052921      0.831373
1          1.000000      -0.304609     0.238865
2          0.000000      0.000000      0.000000
```

4C　估计频率响应的 C 程序

程序用直接估计方法或者用4.5.5节描述的FFT方法计算频率响应。频率响应要估计的系统的 z 变换必须是直接形式或者串联形式，下面用一个例子来清楚地说明这一点。

例 4C.1　求传递函数由下式给出的离散时间系统的频率响应，

$$H(z)=\frac{1-1.6180z^{-1}+z^{-2}}{1-1.5161z^{-1}+0.878z^{-2}}$$

利用

(1) 直接估计方法；

(2) FFT 方法。

假定抽样频率为 500 Hz，分辨率<1 Hz。

解：

为了满足期望的分辨率，程序应用的频率点数 npt 对于 FFT 方法是 512（500/512 = 0.98 Hz），对于直接估计方法是 256，输入数据文件为

```
1
1  -1.5161   0.878
1  -1.618   1
```

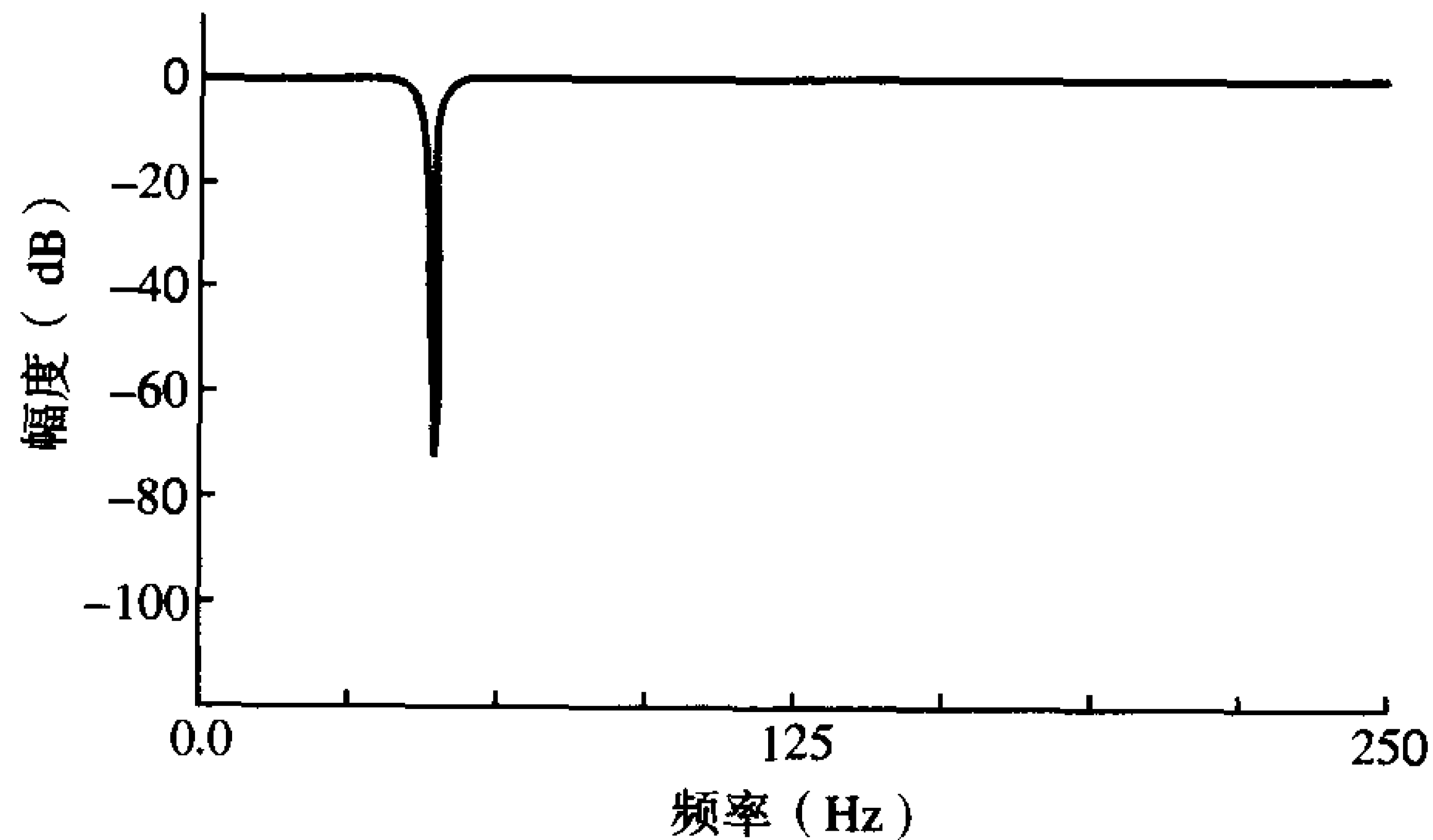
对每一种方法求出频率响应。在两种情况下，响应用 ASCII 格式保存在如下三个文件中：

magn.dat	包含幅度响应，用分贝表示
phase.dat	包含相位响应，用弧度表示
fresp.dat	包含频率响应，用矩阵形式表示

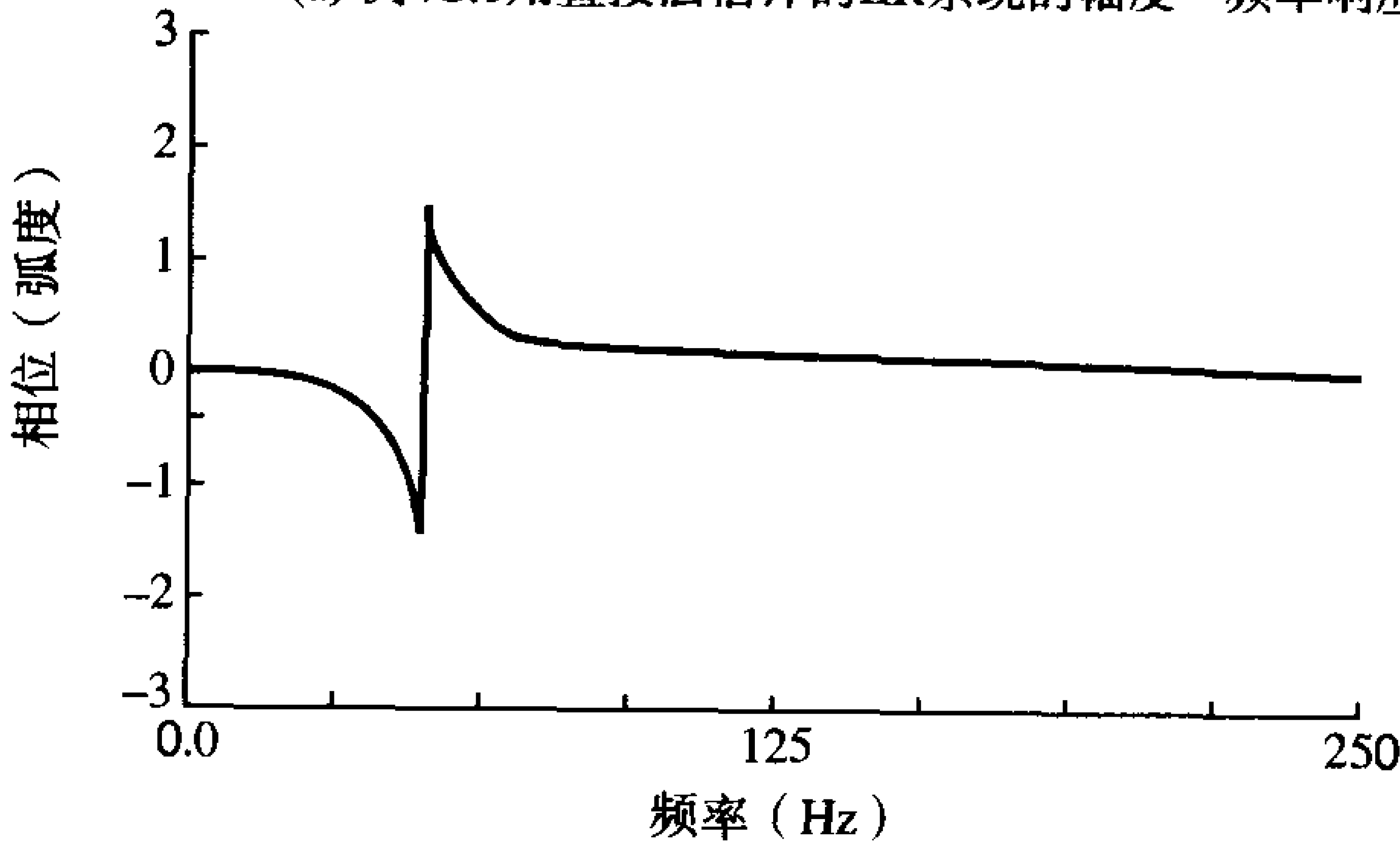
幅度和相位响应的前 10 个值列在表 4C.1 中，直接方法的幅度和相位响应分别在图 4C.1(a)和图 4C.1(b)中给出。

表 4C.1　例 4C.1 中用直接估计或 FFT 方法的幅度和相位响应的前 10 个值

	直接估计		FFT 估计	
k	幅度（dB）	相位（弧度）	幅度（dB）	相位（弧度）
0	0.469 496	0.000 000	0.469 496	0.000 000
1	0.469 39	−0.004 155	0.469 391	−0.004 138
2	0.469 073	−0.008 318	0.469 076	−0.008 286
3	0.468 541	−0.012 500	0.468 549	−0.012 451
4	0.467 791	−0.016 710	0.467 805	−0.016 644
5	0.466 817	−0.020 956	0.466 839	−0.020 873
6	0.465 612	−0.025 249	0.465 643	−0.025 148
7	0.464 165	−0.029 599	0.464 208	−0.029 479
8	0.462 466	−0.034 016	0.462 523	−0.033 876
9	0.460 501	−0.038 511	0.460 574	−0.038 351



(a) 例4C.1用直接法估计的IIR系统的幅度－频率响应



(b) 例4C.1用直接法估计的IIR系统的相位响应

图 4C.1　直接方法的幅度和相位响应

频率响应的程序由五个函数组成，这五个函数保存在如下不同的文件中：

freqres1.c	主函数
fixdata.c	计算幅度和相角
freqd.c	直接频率响应估计
fft.c	按时间抽取的基 -2 FFT 算法
ltilib.c	常用DSP函数集，与附录4B描述的ltilib.c除了不要求.h文件之外是相同的

函数在这里由于没有空间而没有列出，但是在指导手册的 CD 中可以找到（详细情况请参见前言）。

4D　用 MATLAB 的 z 变换运算

在下几节，我们将说明执行各种 z 变换和 z 反变换运算来实现前几节用 C 语言描述的程序的 MATLAB 函数。MATLAB 和 MATLAB 信号处理工具箱提供了快速、便捷地执行 DSP 系统设计与分析的各种 z 变换和 z 反变换运算的工具。

用 MATLAB 进一步说明 z 变换运算的例子在指导手册 *A Practical Guide for MATALB and C Language Implementations of DSP Algorithms* 中可以找到。

4D.1　z 反变换

执行 z 反变换运算的关键的 MATLAB 函数是 deconv 和 residuez。deconv 函数是用来执行幂级数展开法所要求的长除法，residuez 函数是用来求部分分式系数（留数）和 z 变换的极点的。

4D.1.1　用 MATLAB 的幂级数展开法

在幂级数展开法中，关键的运算是多项式除法，MATLAB 函数 deconv 执行反卷积运算。在幂级数展开法中，我们利用这样一个事实：反卷积运算等价于多项式除法。因此，给定一个 z 变换 $X(z)$具有如下形式：

$$X(z) = \frac{b_0 + b_1 z^{-1} + \ldots + b_n z^{-n}}{a_0 + a_1 z^{-1} + \ldots + a_m z^{-m}} = \frac{b(z)}{a(z)}$$

反卷积命令的格式为

```
[q, r] = deconv(b,a)
```

其中 b 和 a 是按 z 的负幂增加的形式分别表示分子和分母多项式 $b(z)$和 $a(z)$。多项式除法的商在矢量 q 中返回，余数包含在 r 中。为了实现幂级数法，连续地应用长除法运算，这取决于求逆运算所要求的点数。

例 4D.1　用幂级数（多项式除法）和 MATLAB 求 z 反变换 $x(n)$的前五项。假定 z 变换 $X(z)$具有下列形式：

$$X(z) = \frac{1 + 2z^{-1} + z^{-2}}{1 - z^{-1} + 0.3561z^{-2}}$$

解：

MATLAB 命令集和答案在下面给出。首先，形成分子和分母多项式系数矢量。为了确保 MATLAB 的正确维数，将加零到系数矢量 b 中，然后用 deconv 命令来计算 z 反变换。

```
»
» b=[1 2 1];
» a=[1 -1 0.3561];
» n=5;
» b=[b zeros(1, n-1)];
» [x, r]=deconv(b,a);
» disp(x)

  1.0000   3.0000   3.6439   2.5756   1.2780
```

因此，$x(0)=1$，$x(1)=3$，$x(2)=3.6439$，$x(4)=2.5756$，$x(5)=1.2780$。

例 4D.2 用幂级数展开法（多项式除法）和MATLAB求下列z反变换的前五个值：

$$X(z)=\frac{N_1(z)N_2(z)N_3(z)}{D_1(z)D_2(z)D_3(z)}$$

其中

$$N_1(z)=1-1.223\,46z^{-1}+z^{-2}$$

$$N_2(z)=1-0.437\,833z^{-1}+z^{-2}$$

$$N_3(z)=1+z^{-1}$$

$$D_1(z)=1-1.433\,509z^{-1}+0.858\,11z^{-2}$$

$$D_2(z)=1-1.293\,601z^{-1}+0.556\,929z^{-2}$$

$$D_3(z)=1-0.612\,159z^{-1}$$

解：

z变换有三对分子和分母多项式，在MATLAB实现中（程序4D.1），首先形成包含多项式系数的矢量。接着用MATLAB函数sos2tf（二阶项比传递函数）将三对多项式转换成具有一对有理多项式的传递函数，即$b(z)/a(z)$：

$$X(z)=\frac{b(z)}{a(z)}=\frac{b_0+b_1z^{-1}+b_2z^{-2}+\ldots+b_mz^{-m}}{a_0+a_1z^{-1}+a_2z^{-2}+\ldots+a_nz^{-n}}$$

deconv函数用来产生z反变换的系数，z反变换的前五项的值为

$$x(0)=1.0000,\ x(1)=4.6915,\ x(2)=11.4246,\ x(3)=19.5863,\ x(4)=27.0284$$

程序 4D.1

```
n = 5;  % number of power series points
N1 = [1 -1.122346 1]; D1 = [1 -1.433509 0.85811];
N2 = [1 1.474597 1]; D2 = [1 -1.293601 0.556929];
N3 = [1 1 0]; D3 = [1 -0.612159 0];
B = [N1; N2; N3]; A = [D1; D2; D3];
[b,a] = sos2tf([B A]);
b = [b zeros(1,n-1)];
[x,r] = deconv(b,a);   %perform long division
disp(x);
```

4D.1.2 用MATLAB的部分分式展开

MATLAB函数residuez可以用来执行两个多项式之比的z变换$X(z)$的部分分式展开，residuez命令的句法为

```
[r, p, k] = residuez(b, a)
```

其中b和a分别是分子和分母多项式$b(z)$和$a(z)$的系数矢量，$b(z)$和$a(z)$是按z的负幂增加的，

$$H(z)=\frac{b(z)}{a(z)}=\frac{b_0+b_1z^{-1}+b_2z^{-2}+\ldots+b_mz^{-m}}{a_0+a_1z^{-1}+a_2z^{-2}+\ldots+a_nz^{-n}}$$

如果$H(z)$的极点不同，它的部分分式展开具有如下形式：

$$\frac{b(z)}{a(z)}=\frac{r_1}{1-p_1z^{-1}}+\ldots+\frac{r_n}{1-p_nz^{-1}}+k_1+k_2z^{-1}+\ldots+k_{m-n-1}z^{-(m-n)}$$

residuez 函数在矢量r中返回有理多项式$b(z)/a(z)$的留数，极点的位置在p中返回，常数项在k中返回。

例 4D.3　求下列z变换的部分分式展开，

$$X(z)=\frac{1+2z^{-1}+z^{-2}}{1-z^{-1}+0.3561z^{-2}}$$

解：

在本例中，多项式已经是正确的形式，所以我们直接应用命令：

```
» [r, p, k] = residuez([1,2,1], [1, -1, 0.3561])
r =
   -0.9041 - 5.9928i
   -0.9041 + 5.9928i
p =
   0.5000 + 0.3257i
   0.5000 - 0.3257i
k = 2.8082
```

因此，用部分分式展开表示的z变换变成

$$X(z)=2.8082+\frac{r_1}{1-p_1z^{-1}}+\frac{r_2}{1-p_2z^{-1}}$$

其中

$$r_1=-0.9041-5.9928\text{j}\qquad r_2=-0.9041+5.9928\text{j}$$
$$p_1=0.5+0.3257\text{j}\qquad p_2=0.5-0.3257\text{j}$$

例 4D.4　用 MATLAB 求下列z变换的部分分式展开：

$$X(z)=\frac{N_1(z)N_2(z)N_3(z)}{D_1(z)D_2(z)D_3(z)}$$

其中

$$N_1(z)=1-1.223\,46z^{-1}+z^{-2}$$
$$N_2(z)=1-0.437\,833z^{-1}+z^{-2}$$
$$N_3(z)=1+z^{-1}$$
$$D_1(z)=1-1.433\,509z^{-1}+0.858\,11z^{-2}$$
$$D_2(z)=1-1.293\,601z^{-1}+0.556\,929z^{-2}$$
$$D_3(z)=1-0.612\,159z^{-1}$$

MATLAB 函数 sos2tf 用来将分子和分母多项式转换成一对多项式$b(z)/a(z)$。residuez 函数用来求部分分式展开。求$X(z)$的部分分式展开的 MATLAB 命令集在程序 4D.2 中给出，运行 MATLAB 程序给出了部分分式系数：

```
r =
  -1.9022 + 4.6797i
  -1.9022 - 4.6797i
  -9.0607 - 13.5515i
```

```
  -9.0607 + 13.5515i
  24.7049

p =
  0.7168 + 0.5868i
  0.7168 - 0.5868i
  0.6468 + 0.3723i
  0.6468 - 0.3723i
  0.6122

k = 1
```

程序 4D.2

```
N1 = [1 -1.122346 1];
N2 = [1 -0.437833 1];
N3 = [1 1 0];
D1 = [1 -1.433509 0.85811];
D2 = [1 -1.293601 0.556929];
D3 = [1 -0.612159 0];
sos = [N1 D1; N2 D2; N3 D3];
[b, a] = sos2tf(sos);
[r, p, k] = residuez(b, a)
```

4D.2　结构之间的转换——串联到并联转换

MATLAB 提供了一组函数，这种函数允许在不同的格式和结构之间进行转换，这些格式和结构在 DSP 中更容易应用。并行和串联结构之间的转换能力是很有用的。

例 4D.5　用 MATLAB 重新做例 4B.4。

执行转换的 MATLAB 命令集在程序 4D.3 中给出。

程序 4D.3

```
nstage=2;
N1 = [1 0.481199 1];
N2 = [1 1.474597 1];
D1 = [1 0.052921 0.83173];
D2 = [1 -0.304609 0.238865];
sos = [N1 D1; N2 D2];
[b, a] = sos2tf(sos);
[c, p, k] = residuez(b, a);
m = length(b);
b0 = b(m)/a(m);
j=1;
for i=1:nstage
        bk(j)=c(j)+c(j+1);
        bk(j+1)=-(c(j)*p(j+1)+c(j+1)*p(j));
        ak(j)=-(p(j)+p(j+1));
        ak(j+1)=p(j)*p(j+1);
        j=j+2;
end
b0
ak
bk
c
p
k
====
cprealization

b0 =
```

```
    5.0334
ck =
    -0.3766 - 0.2460i
    -0.3766 + 0.2460i
     1.4804 - 1.3903i
     1.4804 + 1.3903i

pk =
    -0.0265 + 0.9116i
    -0.0265 - 0.9116i
     0.1523 + 0.4644i
     0.1523 - 0.4644i

ks = 1

b0 = 1.2023

ak = 1.4746     1.0000   0.0529   0.8317

bk = -0.0000   -0.0000   0.4283   0.1683

c =
  -0.0000 + 0.0000i
  -0.0000 - 0.0000i
   0.2141 - 0.0861i
   0.2141 + 0.0861i
```

4D.3　极零图

MATLAB 函数 zplane 允许计算和显示极零图，命令的语句为

```
zplane(b, a)
```

其中 b 和 a 是分子和分母多项式 $b(z)/a(z)$的系数矢量。在这一格式中，命令首先求极点和零点的位置（即 $b(z)$和 $a(z)$的根），然后画 z 平面图。

例 4D.6　离散时间系统的传递函数为

$$H(z) = \frac{1 - 1.6180z^{-1} + z^{-2}}{1 - 1.5161z^{-1} + 0.878z^{-2}}$$

求并且画出极零图。在每种情况下使用 MATLAB，并且假定抽样频率为 500 Hz，频率响应的分辨率 <1 Hz。

解:

MATLAB 命令为

```
b = [1 -1.6180 1];          % form numerator and denominator polynomials
a = [1 -1.5161 0.878];
zplane(b,a)                 % compute and plot the pole-zero diagram
```

极零图如图 4D.1 所示。

如果极点和零点的位置是已知的，这些可以用做 zplane 命令的输入。这时，命令的句法为 zplane(z, p)，其中 z 和 p 是零点和极点。

极点和零点的位置可以直接用 roots 命令求出。对于在极点、零点和传递函数表示之间的转换这是很有用的。

例如，一个 IIR 系统可表示为

$$H(z) = \frac{1 - 1.6180z^{-1} + z^{-2}}{1 - 1.5161z^{-1} + 0.878z^{-2}}$$

滤波器的极点和零点可以用 roots 命令求出：

```
b = [1 -1.618 1];
a = [1 -1.5161 0.878];
zk = roots(b);
pk=roots(a)
```

分子和分母多项式 $b(z)$和 $a(z)$可以用 poly 函数求得：B = poly(zk); A = poly(pk)。

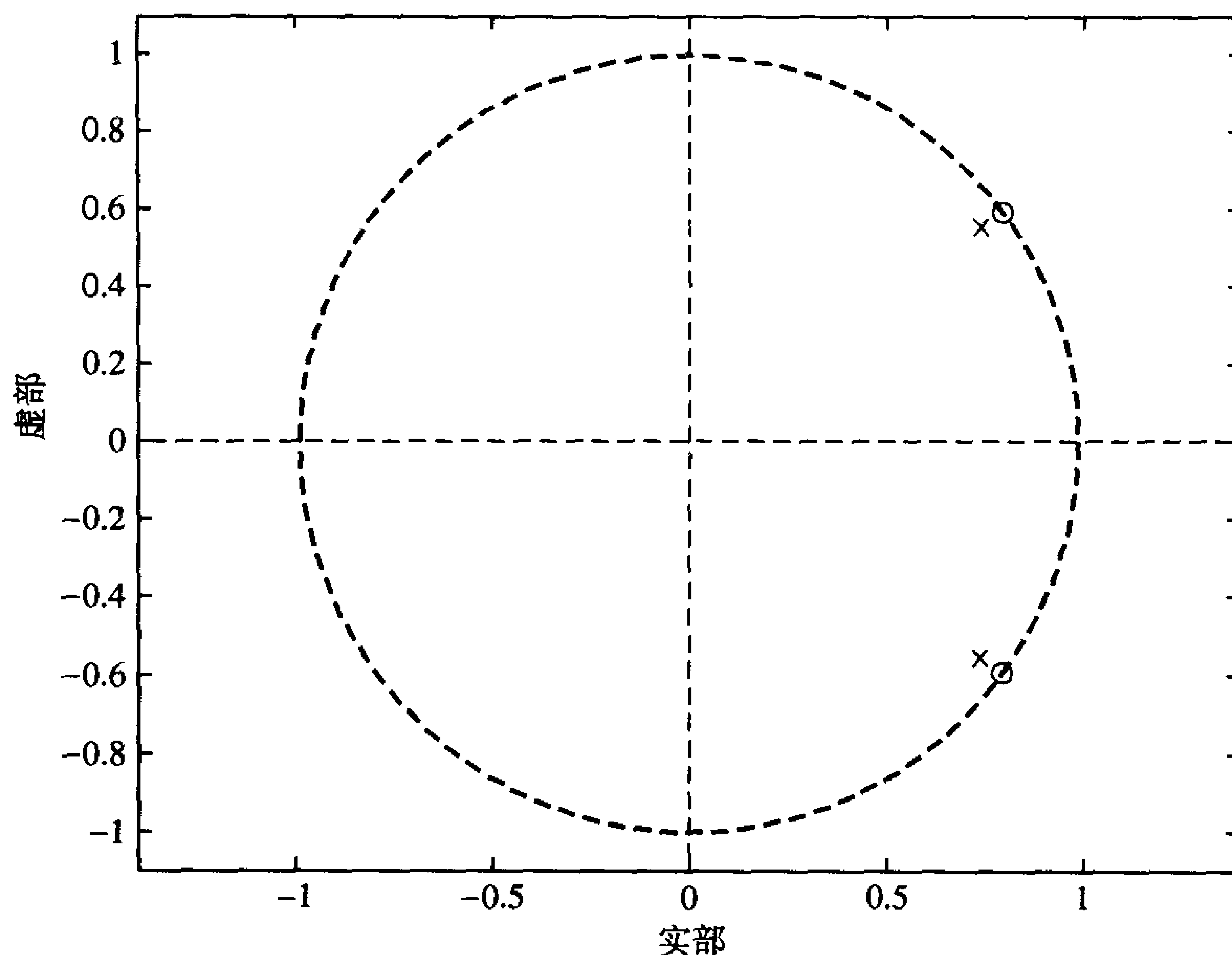


图 4D.1 极零图

4D.4 频率响应估计

信号处理工具箱包含许多计算和显示离散时间系统的有用函数，应用最广泛的是 freqz 函数。给定系统的传递函数为如下形式：

$$X(z) = \frac{b_0 + b_1z^{-1} + ... + b_nz^{-n}}{a_0 + a_1z^{-1} + ... + a_mz^{-m}} = \frac{b(z)}{a(z)}$$

freqz 函数应用了基于 FFT 的方法来计算频率响应，函数有多种格式，有用的格式是[h,f] = freqz (b, a, npt, Fs)，其中变量 b 和 a 是分子和分母多项式的系数矢量，Fs 是抽样频率，npt 是 0 到 $F_s/2$ 之间的频率点数。在 MATLAB 工具箱中，奈奎斯特频率（即 $F_s/2$）是归一化频率单位。利用没有输出变量的 freqz 命令可以自动画出幅度和相位响应。例 4D.7 说明了用 freqz 命令计算离散时间系统频率响应的方法。

例 4D.7 离散时间系统的传递函数为

$$H(z) = \frac{1 - 1.6180z^{-1} + z^{-2}}{1 - 1.5161z^{-1} + 0.878z^{-2}}$$

用 MATLAB 求并且画出系统的频率响应。假定抽样频率为 500 Hz，分辨率 < 1 Hz。

解：

MATLAB 命令是

```
b = [1 -1.6180 1];          % form numerator and denominator coefficient vectors
a = [1 -1.5161 0.878];
freqz(b,a,256,500)          % compute and plot the frequency response
```

离散时间系统的频率响应如图 4D.2 所示。

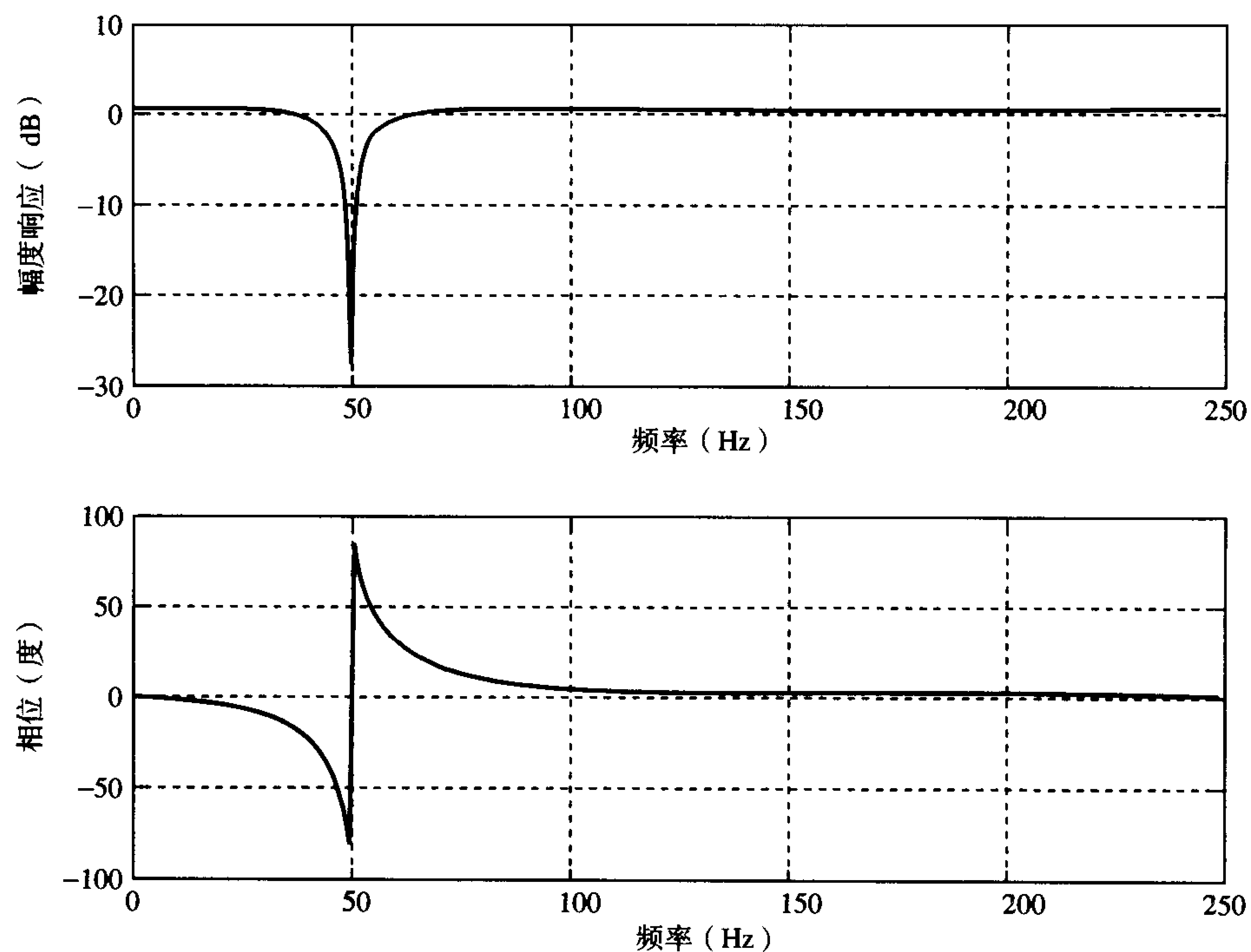


图 4D.2 离散时间系统的频率响应

附录的参考文献

Jury E.I. (1964) *Theory and Applications of the z-transform Method*. New York: Wiley.

Signal Processing Toolbox User's Guide. The Math Works, 1998.

第5章　相关和卷积

本章首先对相关过程的性质进行了描述,接着用一个验证过的关于互相关和自相关的计算例子进行解释。并且通过许多相关应用的例子描述了相关对信号的噪声分量衰减的影响,然后解释了利用FFT的快速相关技术。本章利用类似于相关的方式涵盖了关于卷积的主题。卷积的处理包括循环卷积、线性卷积、快速线性卷积以及处理大量输入数据所必需的分段法(交叠相加,交叠存储)。反卷积过程(deconvolution)也包含在本章里。我们还建立了相关和卷积之间的关系，最后用一节关于实现以及几个验证过的应用例子来结束本章的讨论。

5.1　引言

我们经常需要定量地描述一个过程和另一个过程的相互关系,或者是确定一列数据和另一列数据的相似性。也就是说,要寻找过程或数据间的相关性。相关可以在数学上做出定义以及定量表示。相关在信号处理中占有重要的地位。相关运算出现在一些应用中，例如机器人视觉中的图像处理,卫星遥感(来自不同图像的数据进行比较)，雷达和声呐系统(通过比较发射的和反射的波形来确定距离和位置)，噪声中的信号检测和识别，控制工程(观测输入对输出的影响)，以及脉冲编码调制系统中利用相关检测器的二进制码字的识别，相关在这些领域都有着广泛的应用。另外，相关处理在普通的最小平方估计技术中是作为积分的一部分,在计算波形的平均功率中以及其他的许多领域(例如气候学)中也用到了相关。相关也是卷积过程的一个积分部分。卷积过程实质上也是两个数据序列的相关，其中一个数据序列被翻转过来。这意味着可以用同一个算法来计算相关和卷积，只是卷积时需要把其中的一个序列翻转过来。卷积过程给出了一个对输入滤波的系统的输出。一个记录信号的频谱由信号的频谱和它的窗函数的频谱的卷积组成。

确定一个未知系统的冲激响应称为系统识别。从系统冲激响应和输出信号来确定未知的输入称为反卷积。当冲激响应未知时,未知的输入信号的确定称为盲反卷积。这些重要主题我们将会一一加以描述。

5.2　相关描述

考虑两个数据序列，它们是由对应的两个波形的同时抽样的值组成，如何比较这两个数据序列？如果这两个波形的逐点的变化相似,那么通过取相应的点对的乘积之和,就可以得出它们相关的度量。当考虑两个独立的随机数据序列时，这一方法变得更为可信。在这种情况下，当点对的数目增加时，乘积和将趋向于变为零的很小的随机数。这是因为所有的数字，正的或负的，都以相等的概率发生，以至于乘积对在相加时趋于自我抵消。与此相反，一个有限和值的存在表明了存在一定程度的相关性，一个负的和值代表一个负相关，也就是一个变量增加与另一个变量的减少有关。两个数据序列 $x_1(n)$ 和 $x_2(n)$，每个都包括 N 个数据，它们之间的互相关 $r_{12}(n)$ 可以记为

$$r_{12} = \sum_{n=0}^{N-1} x_1(n)x_2(n)$$

这就是互相关的定义。然而，这样产生的结果依赖于采用的抽样点数，这可以通过除以抽样点的数目 N 做归一化来加以修改。此外，这也可以看做是对乘积和的平均。因此，一个改进的定义是

$$r_{12} = \frac{1}{N}\sum_{n=0}^{N-1} x_1(n)x_2(n)$$

然而，这个定义需要修正才能有用。在某些情况下，尽管两个波形是100%相关，但它却有可能显示为零相关。这是有可能发生的，例如，当两个波形异相（out of phase）时，常常会是这种情形。图5.1用波形对此做了解释。从这个图形我们可以看出，在相关意义下每一个点对的乘积是零，因此它们的相关是零，因为 x_1 和 x_2 中总有一个是零。然而，尽管它们是异相，很显然这两个波形是高度相关的。出现相位差的原因可能是 x_1 为参考信号，而 x_2 是电路的延迟输出。为了克服这样的相位差，需要对其中的一个波形相对于另外一个波形做平移或滞后。通常在相关前向左平移 x_2 与波形对齐。图5.2对此进行了解释，这等价于把 $x_2(n)$ 变成 $x_2(n+j)$，其中 j 表示延时量，它是 x_2 必须向左平移的抽样点数。另外一个等价的方法是把 x_1 向右平移，这时互相关的公式变为

$$\begin{aligned} r_{12}(j) &= \frac{1}{N}\sum_{n=0}^{N-1} x_1(n)x_2(n+j) \\ &= r_{21}(-j) = \frac{1}{N}\sum_{n=0}^{N-1} x_2(n)x_1(n-j) \end{aligned} \tag{5.1}$$

实际上两个波形相关时，它们的相位关系可能并不知道，所以为了确定相关的最大值，需要计算许多不同延时的相关值，那么取最大值为正确的相关值。

例 5.1　下面的例子对 r_{12} 的计算进行了解释。在这个例子里数据序列的点数是 n，序列为 x_1 和 x_2，

n	1	2	3	4	5	6	7	8	9
x_1	4	2	−1	3	−2	−6	−5	4	5
x_2	−4	1	3	7	4	−2	−8	−2	1

$$\begin{aligned} r_{12} = &\frac{1}{9}(4\times -4 + 2\times 1 + -1\times 3 + 3\times 7 + -2\times 4 + -6\times -2 + -5\times -8 + \\ &4\times -2 + 5\times 1) \\ = &5 \end{aligned}$$

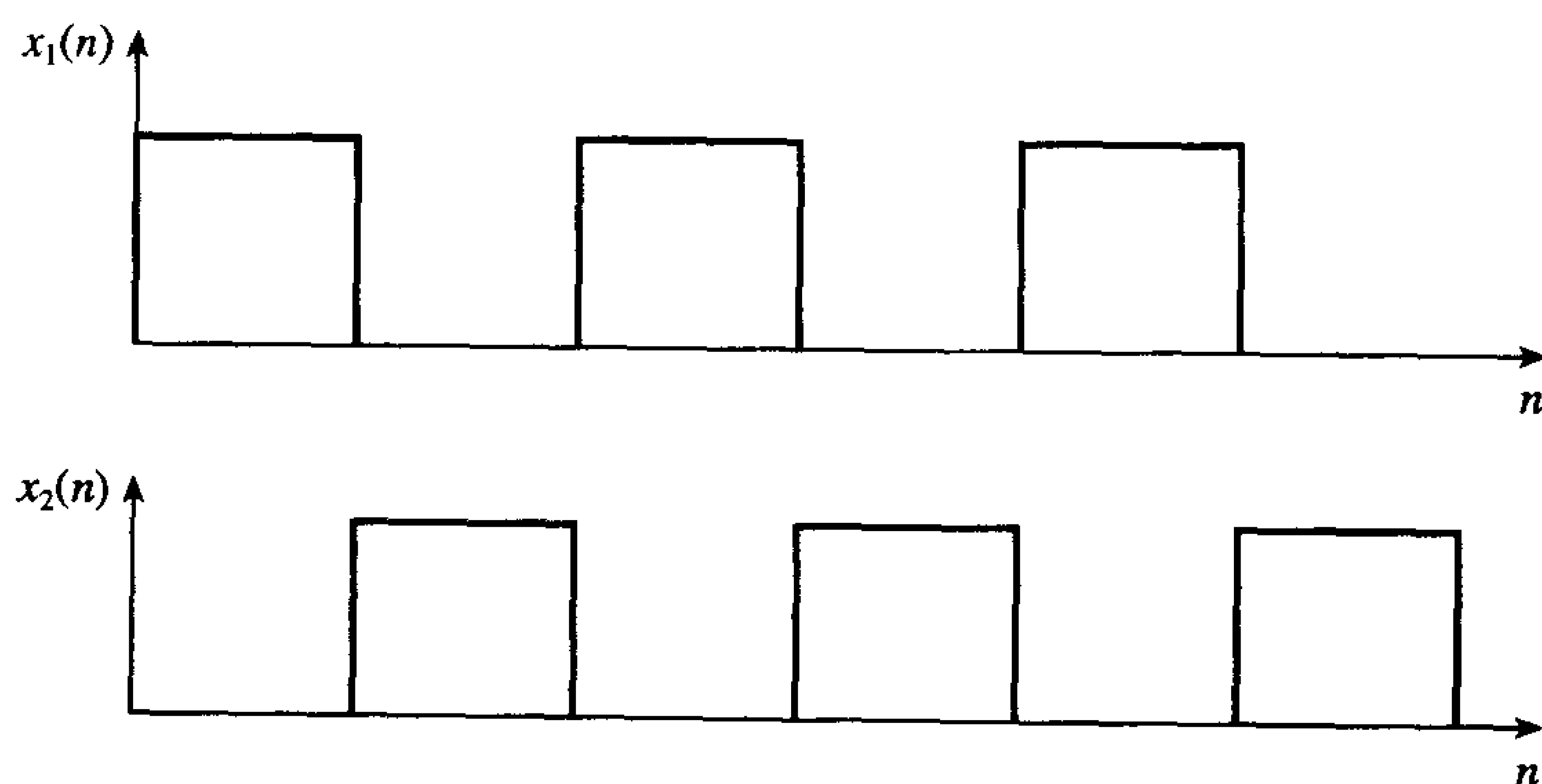


图 5.1　异相但 100% 相关的波形，零延迟时具有零相关

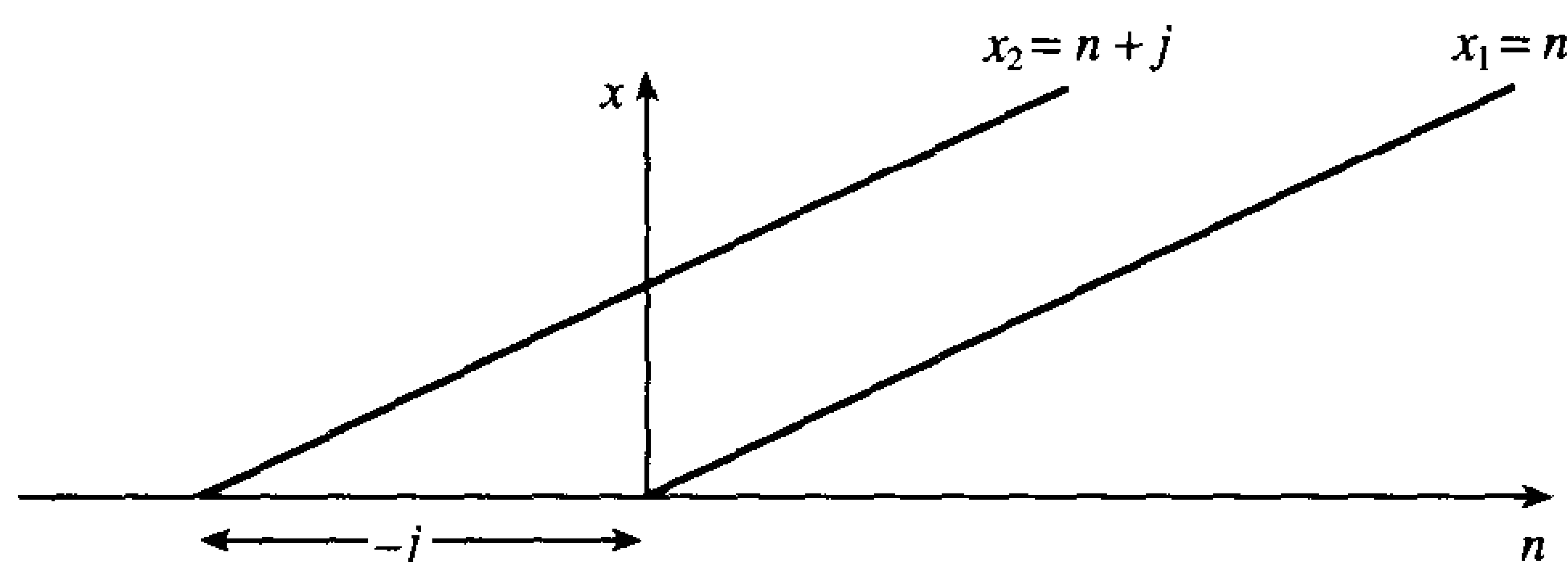


图5.2 波形x_1向左平移j得到的x_2，即$x_2 = x_1 + j$

例5.2 考虑以上的两个序列$x_1(n)$和$x_2(n)$，其中延时$j = 3$，即求$r_{12}(3)$。两个序列变为

n	1	2	3	4	5	6	7	8	9
x_1	4	2	−1	3	−2	−6	−5	4	5
x_2	7	4	−2	−8	−2	−1			

所以

$$r_{12}(3) = \frac{1}{9}(4 \times 7 + 2 \times 4 + -1 \times -2 + 3 \times -8 + -2 \times -2 + -6 \times -1)$$

$$= 2.667$$

当然，在连续的时域内考虑相关也是可能的，某些模拟信号的相关就是以这种方式实现的。在连续域内，$n \to t$，$j \to \tau$，

$$r_{12}(\tau) = \lim_{T \to \infty} \frac{1}{T} \int_{-T/2}^{T/2} x_1(t) x_2(t + \tau)\, dt \tag{5.2}$$

然而，如果$x_1(t)$和$x_2(t)$是周期为T_0的周期函数，5.2式可以简化为

$$r_{12}(\tau) = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} x_1(t) x_2(t + \tau)\, dt \tag{5.3}$$

如果波形是有限能量波形，例如是非周期的脉冲类型的波形，那么，在时间间隔T上当$T \to \infty$时计算平均值并不是可取的，因为这时$1/T \to 0$，$r_{12}(\tau)$总是很小并趋于零。对于这种情形，原则上采用5.4式：

$$r_{12}(\tau) = \int_{-\infty}^{\infty} x_1(t) x_2(t + \tau)\, dt \tag{5.4}$$

实际上，处理的是一个有限记录长度波形，所以将应用5.5式或者5.1式：

$$r_{12}(\tau) = \frac{1}{T} \int_{0}^{T} x_1(t) x_2(t + \tau)\, dt \tag{5.5}$$

有限长度数据的互相关还存在另外一个困难。这可以从上面的例子看出，在这个例子里求得$r_{12}(3) = 2.667$，当x_2向左做平移时，波形不再重叠，序列末端的数据不再形成乘积对。这称为“尾端效应”（end effect）。在这个例子里，当延迟为3时，点对的数目从9下降到6，结果是$r_{12}(j)$随j的增加而线性减少，造成了$r_{12}(j)$的有争议的结果。一个可能的解决方法是使其中的一个序列的长度变为要求的相关长度的2倍。通过记录更多的数据就可以达到这一点。或者，如果其中的一个序列是周期的，通过重复这个序列（注意使两端匹配）也可以达到。另外一个可能的方法是对所有计算值做校正。图5.3说明了由于尾端效应，$r_{12}(j)$如何随着j的增大而减少，而$r_{12}(j)$的实际变化并没有包括

在内。当$j=0$时，可以算得$r_{12}(j)=r_{12}(0)$。当$j=N$时，$r_{12}(N)=0$，因为波形不再重叠。延时j介乎两者之间时，对于某个延迟j，$r_{12}(j)$的真实值为$r_{12}(j)_{\text{true}}$，而由尾端效应引起的实际值是$r_{12}(j)$。那么，从图上可以得出

$$\frac{r_{12}(j)_{\text{true}}-r_{12}(j)}{j}=\frac{r_{12}(0)}{N}$$

因此，

$$r_{12}(j)_{\text{true}}=r_{12}(j)+\frac{j}{N}r_{12}(0) \tag{5.6}$$

因此，考虑到尾端效应，互相关的计算值很容易通过在$r_{12}(j)$上加上$jr_{12}(0)/N$来进行校正。

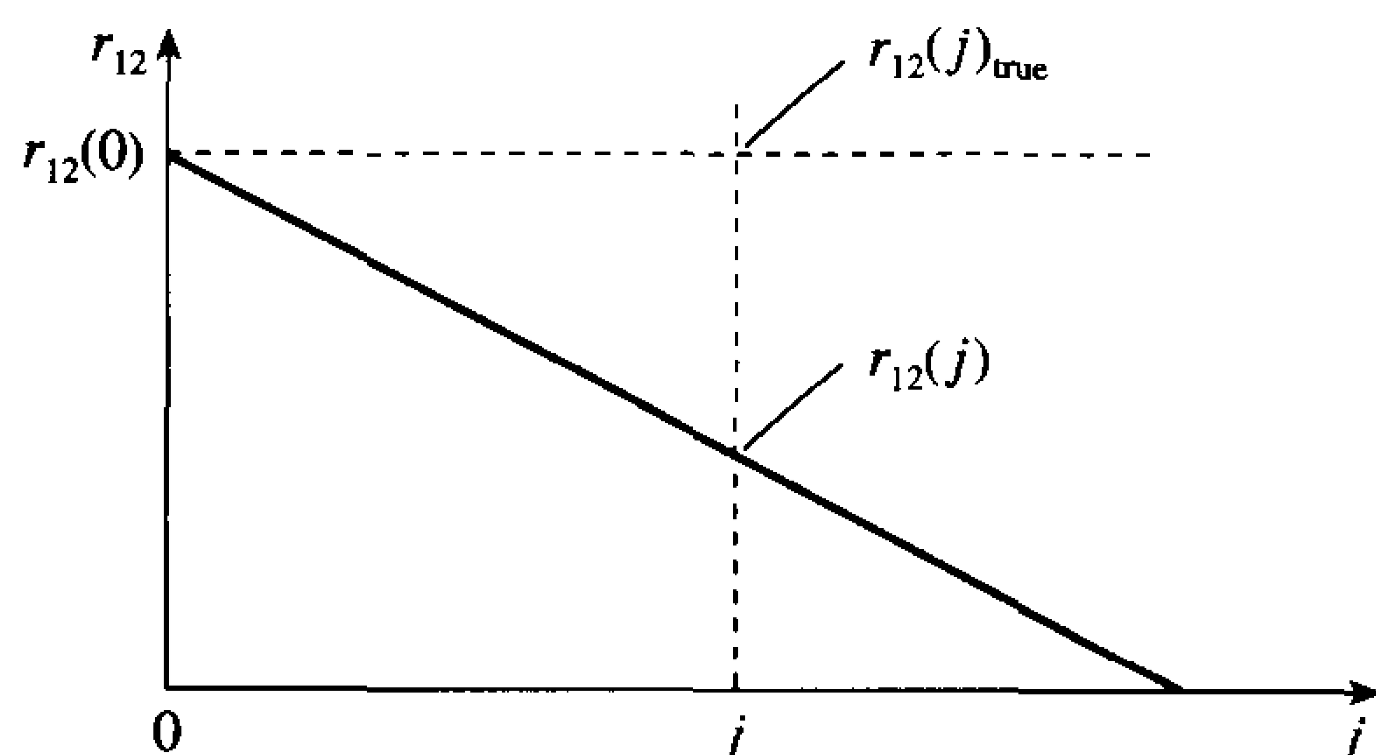


图 5.3　尾端效应对互相关$r_{12}(j)$的影响

依据上面的公式计算出来的互相关值与数据的绝对值有关。根据-1 ~ +1之间的固定比例来度量互相关值是必需的。这可以通过用一个数做归一化来实现，这个数是一个与数据的能量有关的量。例如，考虑两对波形$x_1(n)$、$x_2(n)$和$x_3(n)$、$x_4(n)$。数据值如下表所示：

n	0	1	2	3	4	5	6	7	8
$x_1(n)$	0	3	5	5	5	2	0.5	0.25	0
$x_2(n)$	1	1	1	1	1	0	0	0	0
$x_3(n)$	0	9	15	15	15	6	1.5	0.75	0
$x_4(n)$	2	2	2	2	2	0	0	0	0

从图 5.4 中可以看出，波形$x_1(n)$和$x_3(n)$很像，仅在幅度上有差别。同样$x_2(n)$和$x_4(n)$也是如此，因此，$x_1(n)$与$x_2(n)$的相关与$x_3(n)$与$x_4(n)$的相关相同。然而，互相关$r_{12}(1)$和$r_{34}(1)$分别是 1.47 和 8.83。它们不相同是因为它们依赖于数据的绝对值。对于这种情况，通过使用如下因子对互相关$r_{12}(j)$进行归一化就可以修正这种情况，

$$\left[\frac{1}{N}\sum_{n=0}^{N-1}x_1^2(n)\times\frac{1}{N}\sum_{n=0}^{N-1}x_2^2(n)\right]^{1/2}=\frac{1}{N}\left[\sum_{n=0}^{N-1}x_1^2(n)\sum_{n=0}^{N-1}x_2^2(n)\right]^{1/2} \tag{5.7}$$

对$r_{34}(j)$也采用类似的方法。那么$r_{12}(j)$的归一化表达式变成

$$\rho_{12}(j)=\frac{r_{12}(j)}{\frac{1}{N}\left[\sum_{n=0}^{N-1}x_1^2(n)\sum_{n=0}^{N-1}x_2^2(n)\right]^{1/2}} \tag{5.8}$$

$\rho_{12}(j)$称之为互相关系数。它的值总是位于-1 和+1 之间。+1 意味着在相同的意义下 100%的相关；-1 意味着在相反的意义下 100%相关，例如反相信号。0 值表示零相关。这意味着信号是完全不相关的。例如，如果两个波形中有一个是完全随机的，那么就是这种情形。小的$\rho_{12}(j)$值意味着相关性很低。在上面的说明中，$r_{12}(j)$的归一化因子是

$$\frac{1}{N}\left[\sum_{n=0}^{N-1} x_1^2(n) \sum_{n=0}^{N-1} x_2^2(n)\right]^{1/2} = \frac{1}{9}(88.31\times 6)^{1/2} = 2.56$$

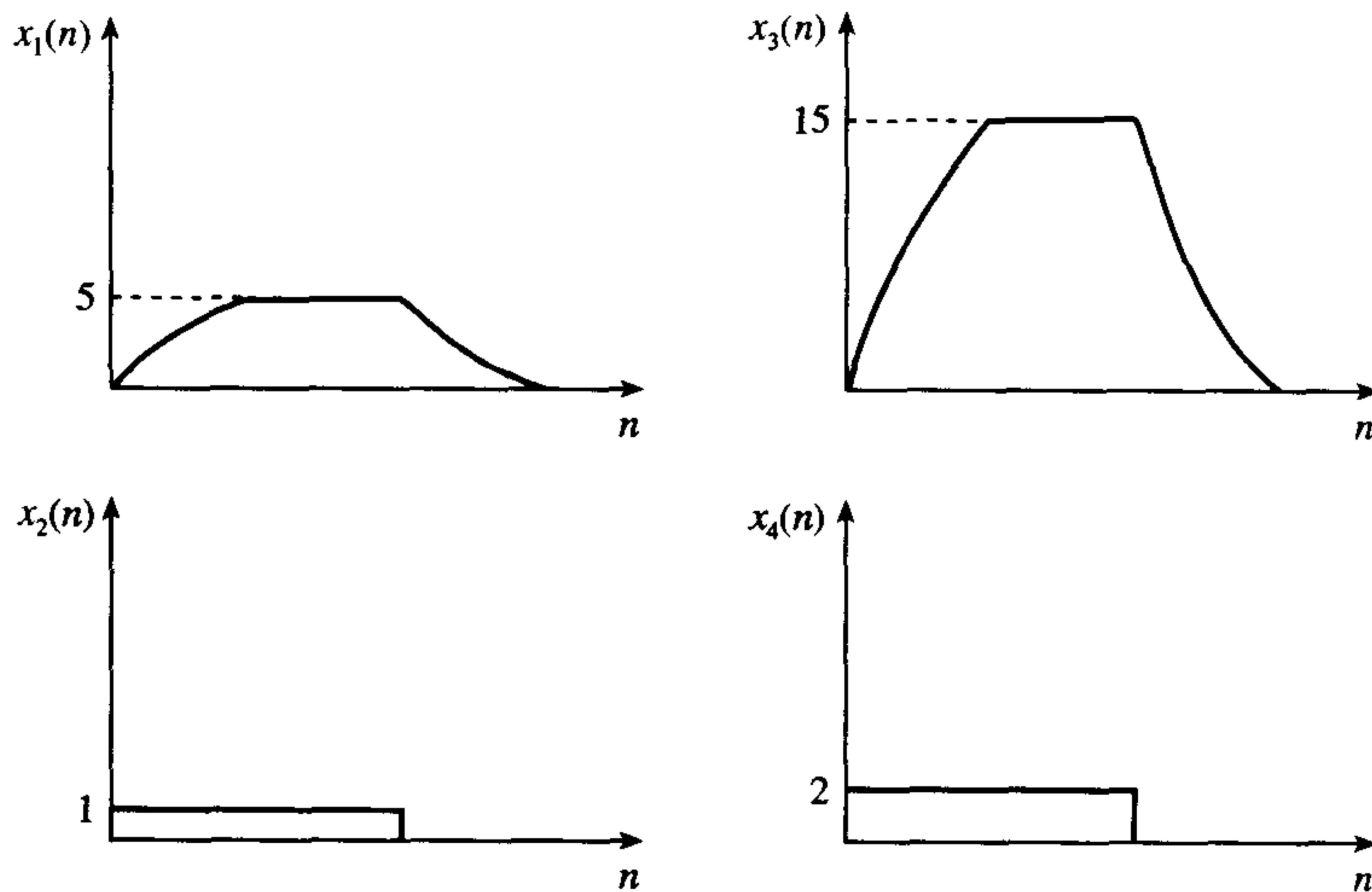


图 5.4 不同幅度的波形对$\{x_1(n), x_2(n)\}$, $\{x_3(n), x_4(n)\}$，但具有相等的互相关

而对于$r_{34}(j)$，它是

$$\frac{1}{N}\left[\sum_{n=0}^{N-1} x_3^2(n) \sum_{n=0}^{N-1} x_4^2(n)\right]^{1/2} = \frac{1}{9}(794.8\times 24)^{1/2} = 15.35$$

因此，

$$\rho_{12}(1) = \frac{r_{12}(1)}{2.56} = \frac{1.47}{2.56} = 0.57$$

以及

$$\rho_{34}(1) = \frac{r_{34}(1)}{15.34} = \frac{8.83}{15.35} = 0.58$$

现在$\rho_{12}(1) = \rho_{34}(1)$，这表明了归一化过程确实允许互相关的比较，而这种比较与数据的绝对值无关。

当$x_1(n) = x_2(n)$时会出现一种特殊的情形，也就是波形和它自身求互相关。这个过程称为自相关。一个波形的自相关是这样给定的：

$$r_{11}(j) = \frac{1}{N}\sum_{n=0}^{N-1} x_1(n)x_1(n+j)$$

自相关函数有一个非常有用的性质，即

$$r_{11}(0) = \frac{1}{N}\sum_{n=0}^{N-1} x_1^2(n) = S$$

其中S是波形归一化的能量。这为计算一个信号的能量提供了一种方法。如果这个波形是完全随机的，例如对应于一个电子系统的高斯白噪声，那么这个自相关在零延时处将达到它的峰值，延时超过一个单位时将衰减到一个零附近的随机波动（参见图5.5）。这构成了对随机波形的检验方法。对这个主题更详细的探讨请参见5.2.1节。另一个性质是

$$r_{11}(0) \geqslant r_{11}(j)$$

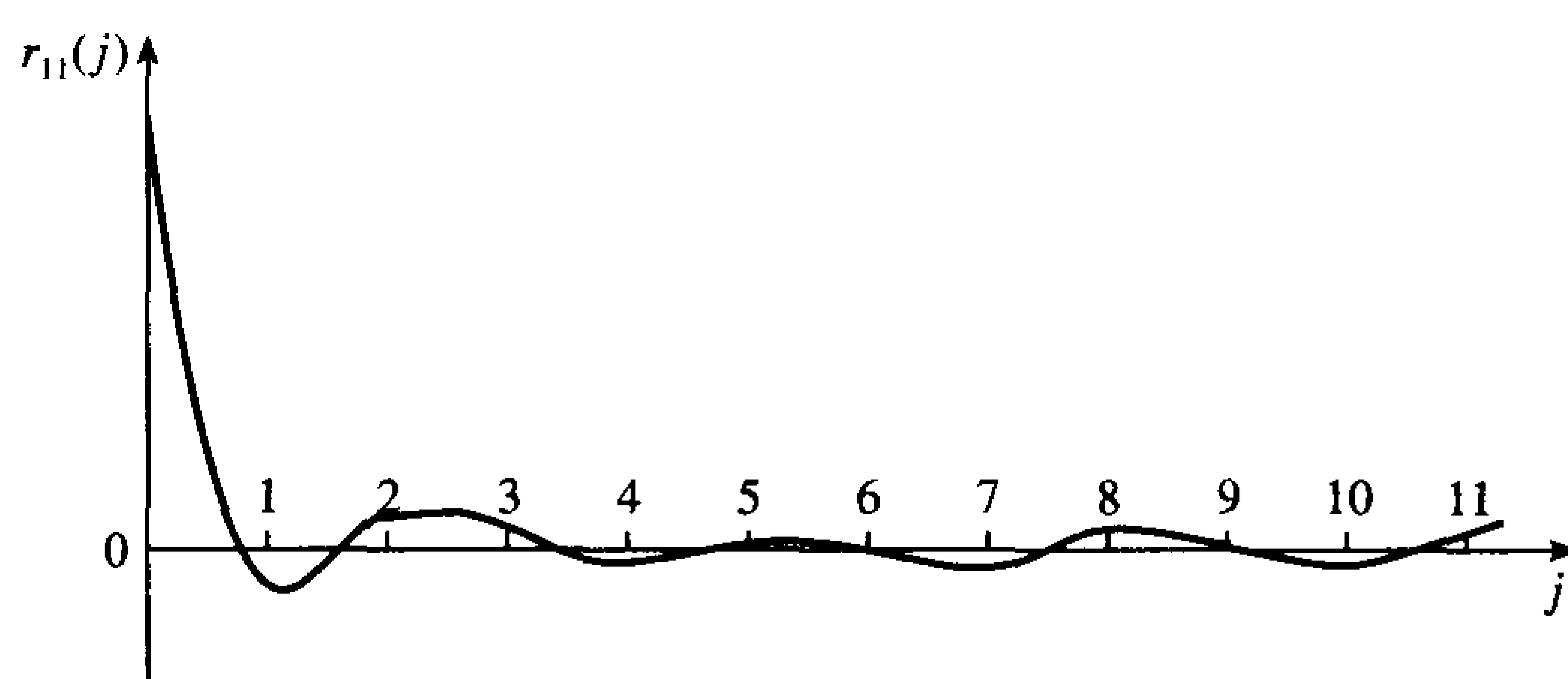


图 5.5 随机波形的自相关函数

5.2.1 互相关和自相关

当两个长度不等的、周期的序列求互相关时必须小心。这是因为相关的结果将是以较短的序列为周期的循环函数。这个结果不能表现较长的那个序列的完整的周期性，因而是不正确的。这可以用计算序列 $a = \{4, 3, 1, 6\}$ 和 $b = \{5, 2, 3\}$ 的互相关 $r_{ab}(j)$ 来说明。序列 b 放置在序列 a 的下面，对 b 序列随后各行依次向左平移一个延迟，互相关的值出现在右边的最后一列。

序列				延迟	$r_{ab}(j)$	
4	3	1	6			
3	5	2	3	0	47	
5	2	3	5	1	59	
2	3	5	2	2	34	
3	5	2	3	3	47	$r_{ab}(j)$重复
5	2	3	5	4	59	
等等						

结果表明 $r_{ab}(j)$ 是循环的，每三个延时重复一次，即 $r_{ab}(j)$ 具有和较短的系列 b 一样的周期。这个过程称为循环相关。a 的每一个值与 b 的每一个值相乘，为了得到一个正确的值，b 的所有元素按如下所示的形式在 a 的每一个值下面依次平移：

```
          4 3 1 6
             5 2 3
            5 2 3
           5 2 3
          5 2 3
         5 2 3
        5 2 3
       5 2 3
```

可以看出在重复 b 序列之前要求 6 个延迟，序列长度是 4 和 3，则所需要的延迟的数目是 $4+3-1=6$。这表示了求两个长度为 N_1 和 N_2 的周期性序列的线性互相关的一般规则：给每个序列补零，使每个序列的长度变成 N_1+N_2-1。也就是给长度为 N_1 的序列加 N_2-1 个零，给长度为 N_2 的序列加 N_1-1 个零。下面用前面给出的序列 a、b 做个示范：

序列						延迟	$r_{ab}(j)$
4	3	1	6	0	0		
5	2	3	0	0	0	0	29

2	3	0	0	0	5	1	17	
3	0	0	0	5	2	2	12	
0	0	0	5	2	3	3	30	
0	0	5	2	3	0	4	17	
0	5	2	3	0	0	5	35	
5	2	3	0	0	0	6	29	$r_{ab}(j)$重复

等等

因此，要求的 a 和 b 的线性互相关为

$$r_{ab}(j) = \{29, 17, 12, 30, 17, 35\}$$

迄今为止，我们举的互相关的例子都是假设为数字化的数据，但是当波形的解析表达式能写出来时（包括波形要求分段表示的情况），互相关也可以解析地进行。实际上在模拟电路的应用中，解析法对互相关的影响是相同的，下面给出了一个解析的互相关求解的例子。

例 5.3 求如图 5.6 所示的波形 $v_1(t)$和 $v_2(t)$的互相关 $r_{12}(-\tau)$。

通过把波形分段成一个个直线部分，可以很容易把波形表示成解析形式。仅需要在波形的一个周期 T 上求互相关，因为 $r_{12}(-\tau)$是 τ 的周期函数，周期为 T。因此，当 $0 \leqslant t \leqslant T$ 时 $v_1(t) = t/T$，当 $0 \leqslant t \leqslant T/2$ 时 $v_2(t) = 1.0$，而当 $T/2 \leqslant t \leqslant T$ 时 $v_2(t) = -1.0$。要求就是要得到 $r_{12}(-\tau)$的表达式，即矩形波形 $v_2(t)$相对于 $v_1(t)$要向右平移。对 $0 \leqslant \tau \leqslant T/2$ 的情形在图 5.7 进行了描述。图中表明 $v_1(t)$要与 $v_2(t)$三个连续的部分相乘，$v_2(t)$这三个连续的部分的值分别为 −1、1 和 −1。图 5.8 给出了 $T/2 \leqslant \tau \leqslant T$ 的情形，但 $v_2(t)$的值变成了 1、−1 和 +1，这意味着在 $\tau = T/2$ 存在两部分必须匹配的解。

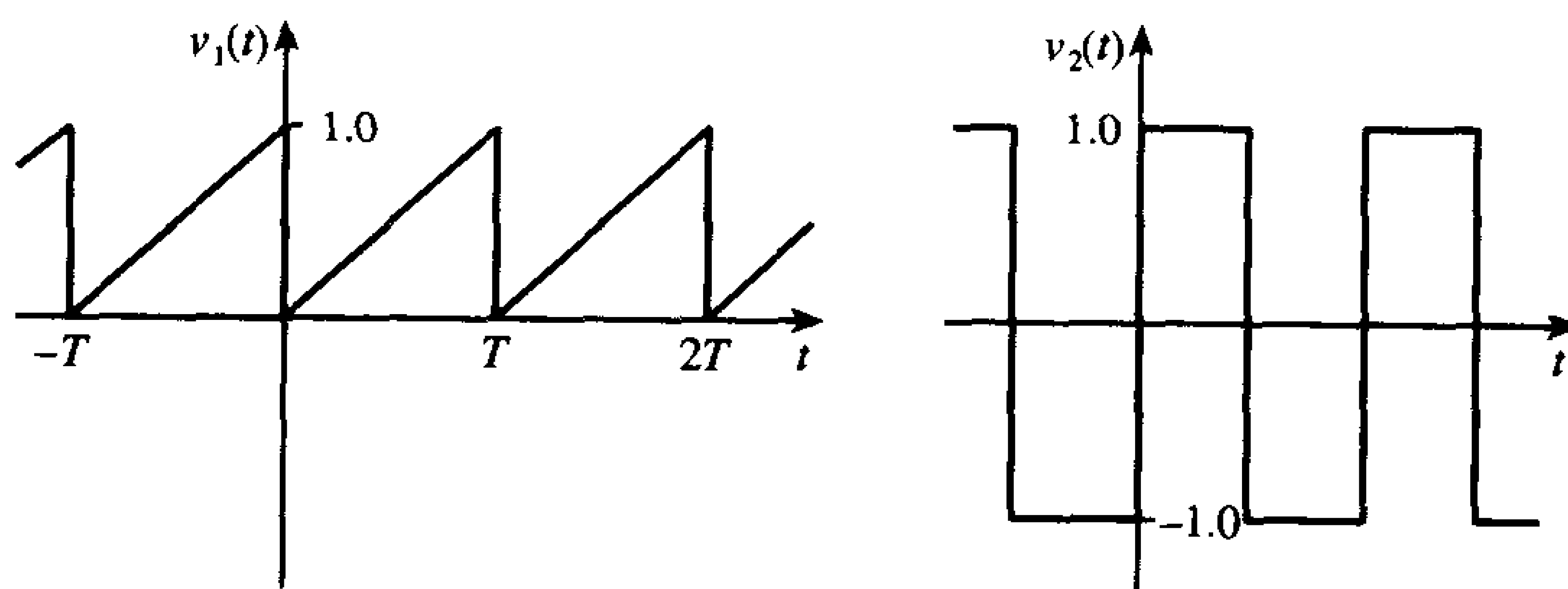


图 5.6　互相关例子的波形 $v_1(t)$和 $v_2(t)$

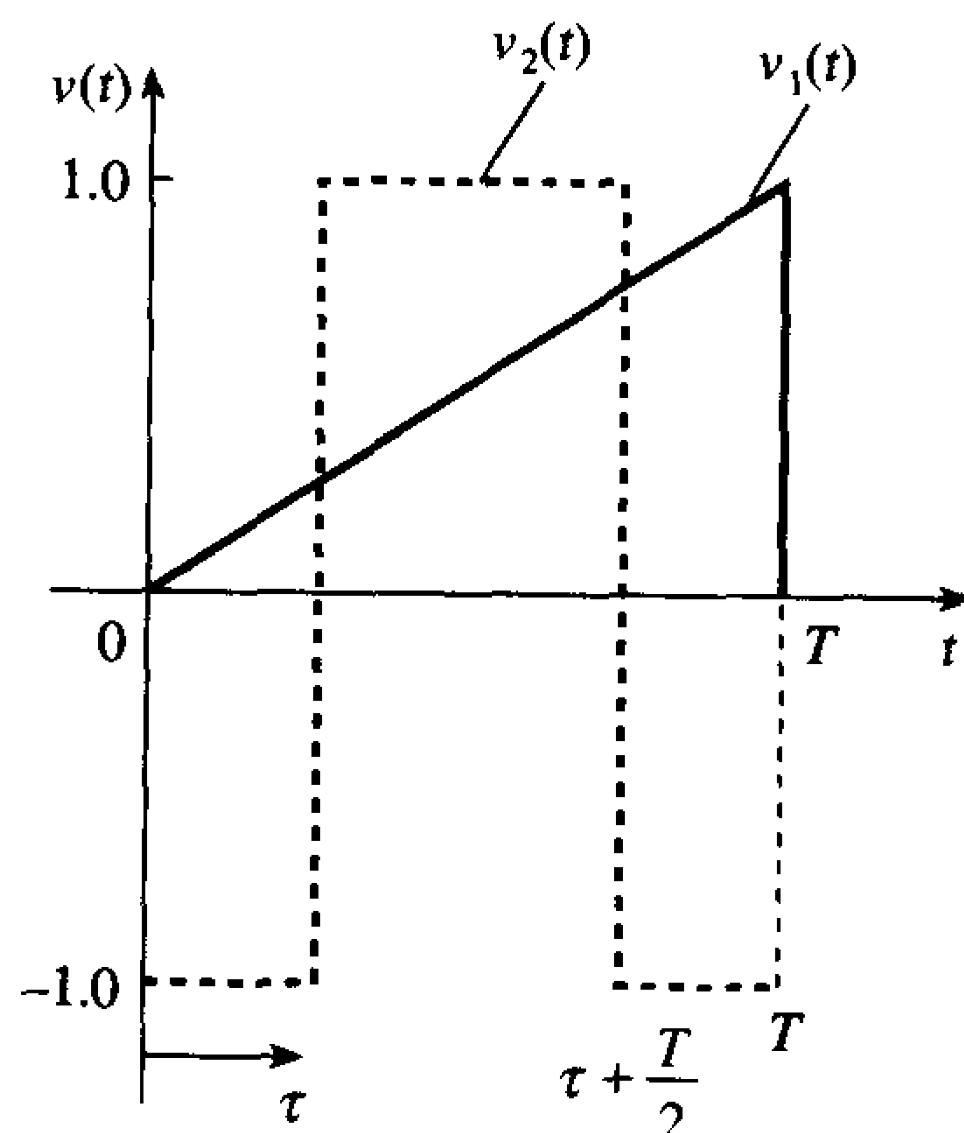


图 5.7　$0 \leqslant \tau \leqslant T$ 时 $v_2(t)$的分段表示

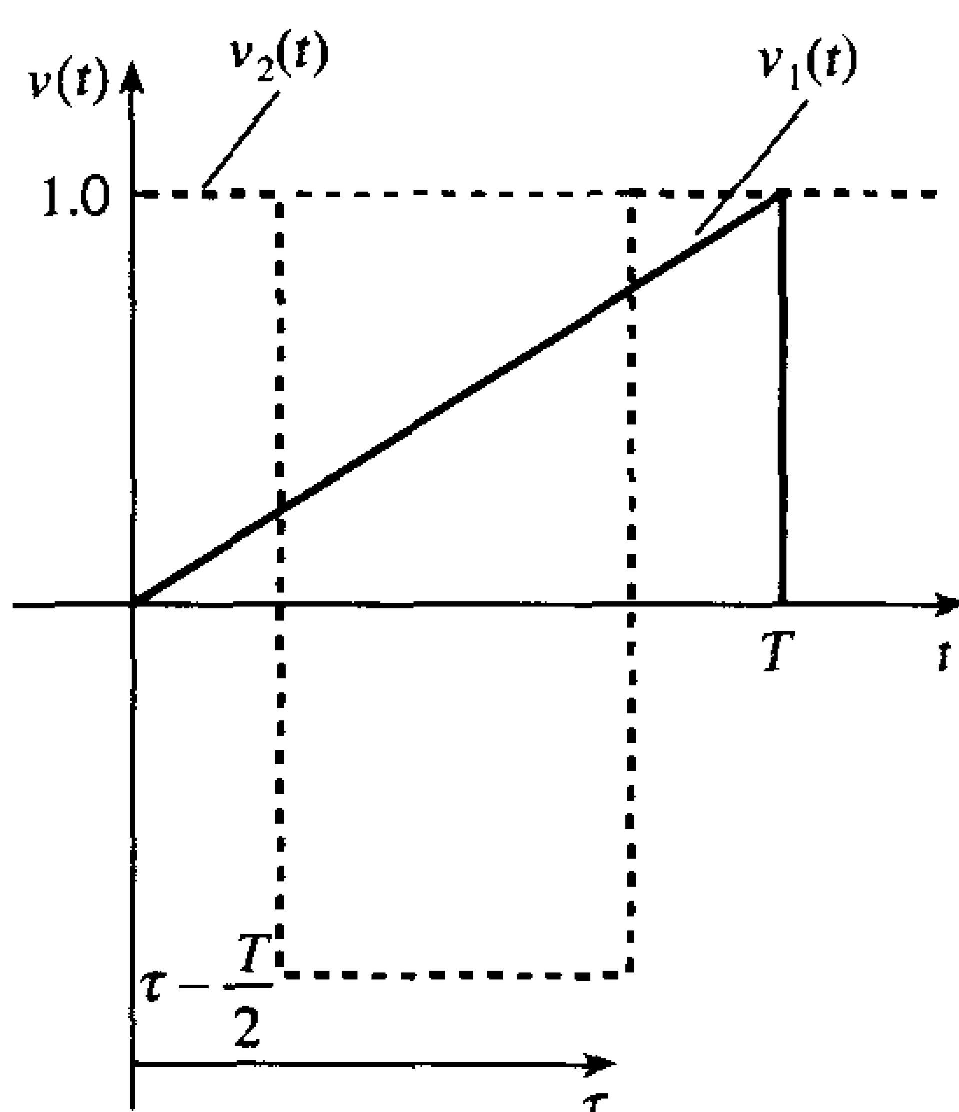


图 5.8　$T/2 \leqslant \tau \leqslant T$ 时 $v_2(t)$ 的分段表示

参考图 5.7，互相关被分成三个部分，其中边界分别是 $t=\tau$、$t=\tau+T/2$ 和 $t=T$。因此，

$$
\begin{aligned}
r_{12}(-\tau) &= \frac{1}{T}\int_0^T v_1(t)v_2(t-\tau)\,\mathrm{d}t \\
&= \frac{1}{T}\int_0^\tau \frac{t}{T}(-1)\,\mathrm{d}t + \frac{1}{T}\int_\tau^{\tau+T/2}\frac{t}{T}(1)\,\mathrm{d}t + \frac{1}{T}\int_{\tau+T/2}^{T}\frac{t}{T}(-1)\,\mathrm{d}t \\
&= \frac{-1}{T^2}\left[\frac{t^2}{2}\right]_0^\tau + \frac{1}{T^2}\left[\frac{t^2}{2}\right]_\tau^{\tau+T/2} - \frac{1}{T^2}\left[\frac{t^2}{2}\right]_{\tau+T/2}^{T}
\end{aligned}
\tag{5.9}
$$

$$r_{12}(-\tau) = -\frac{1}{4}+\frac{\tau}{T} \quad \text{for } 0\leqslant\tau\leqslant\frac{T}{2}$$

当 $T/2 \leqslant \tau \leqslant T$ 时，参考图 5.8，可以看出：

$$
\begin{aligned}
r_{12}(-\tau) &= \frac{1}{T}\int_0^{\tau-T/2}\frac{t}{T}(1)\,\mathrm{d}t + \frac{1}{T}\int_{\tau-T/2}^{\tau}\frac{t}{T}(-1)\,\mathrm{d}t + \frac{1}{T}\int_\tau^T\frac{t}{T}(1)\,\mathrm{d}t \\
r_{12}(-\tau) &= \frac{3}{4}-\frac{\tau}{T}\quad ,\quad \frac{T}{2}\leqslant\tau\leqslant T
\end{aligned}
\tag{5.10}
$$

把 $\tau=T/2$ 代入 5.9 式和 5.10 式，两种情况下都得出 $r_{12}(-\tau)=1/4$，可以确信两个函数进行了正确的匹配。图 5.9 画出了 $0\leqslant\tau\leqslant T$ 时 $r_{12}(-\tau)$ 与 τ 的图。

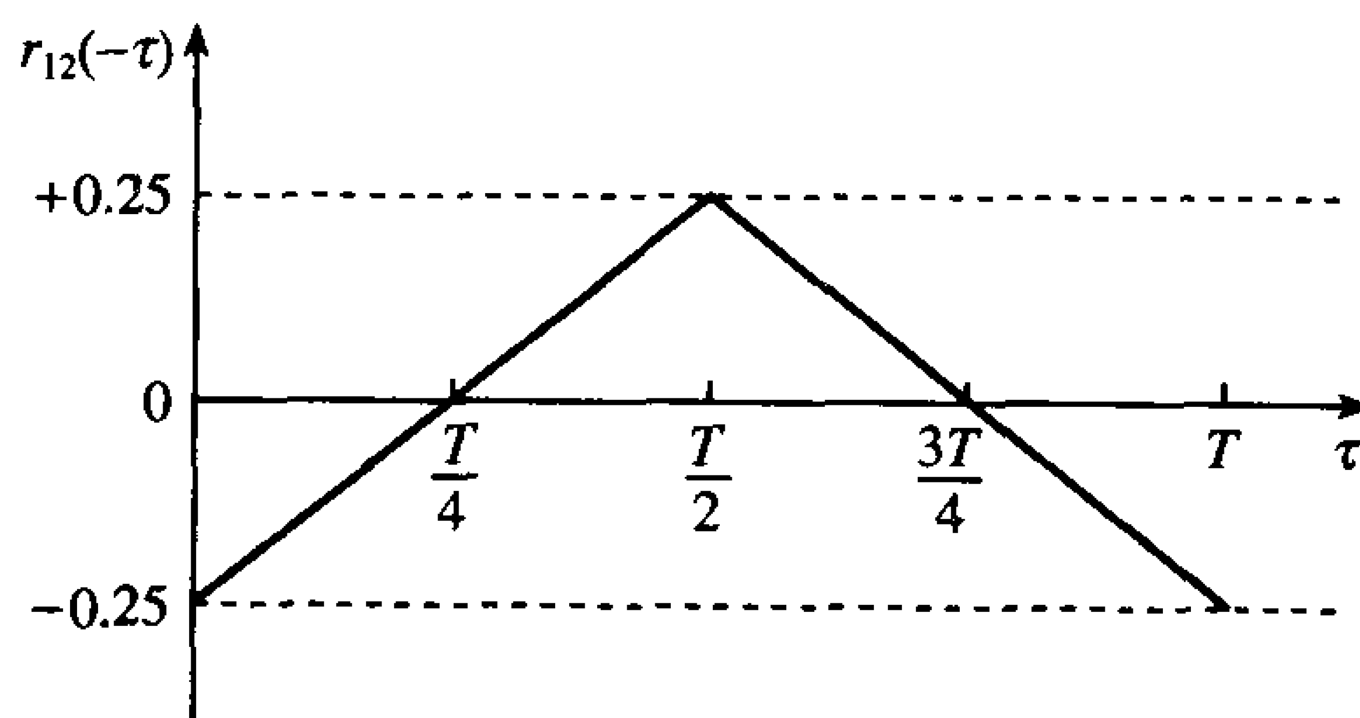


图 5.9　$r_{12}(-\tau)$ 作为 τ 的函数

考虑在相关计算中使用有限长度数据所带来的后果是有益的，换句话说，使用5.5式（其中T是有限的）来代替5.2式会带来什么样的影响?

这个问题可以通过考虑信号的一个正弦傅里叶谐波分量来回答。5.2式在$T \geqslant T_p$时将给出正确的自相关，其中T_p是正弦的周期。这样，

$$r_{11}(\tau) = \lim_{T\to\infty} \frac{1}{2T}\int_{-T}^{T} A\sin(\omega t)A\sin(\omega t+\tau)\,dt$$
$$= \lim_{T\to\infty} \frac{A^2}{2}\left[\cos(\omega\tau) - \frac{\cos(\omega T)}{2\omega T}\sin(\omega\tau)\right] \tag{5.11}$$

检查这个等式可以看出，当$T\to\infty$时括号里的第二项$\to 0$，所以当$T\neq\infty$时，它代表一个误差。$\cos(\omega T)$项代表周期误差的影响，而$1/2\omega T$给出了误差的趋势。因而，就相关长度T而言，序列越短误差越大，波形的低频分量也是最大的。误差也是τ的周期函数。

当$\omega T = [(2n+1)/2]\pi$时，$\cos(\omega T)$项给出了最小误差。由于$\omega = 2\pi/T_p$，且要寻找大的T值，这对应于

$$T \geqslant (2n+1)\frac{T_p}{4} \tag{5.12}$$

当$\omega\tau = m\pi$时，其中m是整数，$\sin(\omega\tau)$最小。因此，

$$\tau = \frac{m}{2}T_p \tag{5.13}$$

现在做一些合理的假设是有必要的。假定大T的条件由$n \geqslant 10$满足。那么，$T \geqslant nT_p/2$，即

$$T \geqslant 5T_p \tag{5.14}$$

从5.13式，对于最低的频率分量而言（$m = 1$），τ的最大允许值满足

$$\tau < T_p \tag{5.15}$$

组合5.14式和5.15式，

$$\tau \leqslant T/5$$

这意味着求波形的自相关时，由于有限数据长度而引起得误差可以通过如下方式减到最小:

(1) 确保$T \geqslant 5T_p$，其中T_p对应于感兴趣的最低频率分量;
(2) 数据的重叠不超过它们长度的20%。

因此，如果要对带宽在300 Hz到3.4 kHz且以40 kHz抽样的电话语音信号求相关，那么$T_p = 1/300 = 3.3\times10^{-3}$ s。可接受的最小数据长度为$5\times3.3\times10^{-3}\text{s} = 16.7$ ms，最大的相关平移是3.33 ms，或者说133个数据点。

图5.10给出了$\rho_{11}(j)$的图，即随机波形（如白噪声）的自相关系数。可以证明，$r_{11}(j)$的期望值为$E[r_{11}(j)]\approx -1/N$（Chatfield, 1980），其中N是数据点数，它的方差是$\text{var}[r_{11}(j)]\approx 1/N$。在图中显示的$-1/N$的期望值其95%是在$-1/N$的置信限度内，即$\pm2/N^{1/2}$。落在这些置信限度之外的$r_{11}(j)$的值可能是具有显著性意义的，即它们可能指出了波形不是真正随机的。然而，应该注意的是，甚至当波形是完全随机的，20个点中也有一个位于这些置信限度以外。对于一个随机波形，$r_{11}(j)$在一个或两个延迟里95%落在置信限度之内。要确定一个波形是随机需要一定的经验和技巧。例如，对数据进行预白化可能是很可取的（Jenkins and Watts, 1968）。

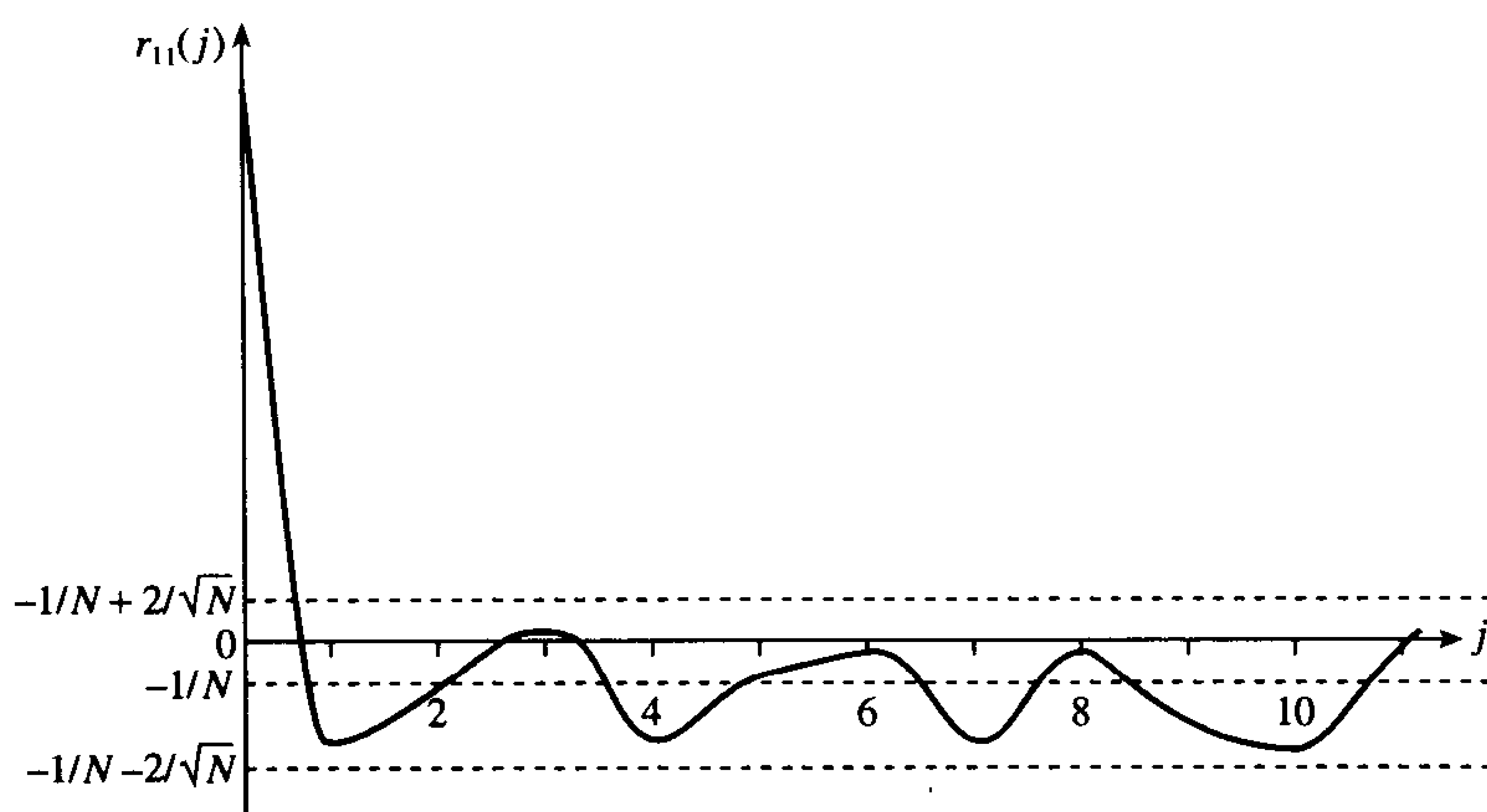


图 5.10　随机波形的自相关系数

一个周期波形的自相关函数也是周期的。这很容易证明。周期为 T 的周期波形满足：

$$x(t) = x(t + nT)$$

所以

$$r_{11}(\tau) = \lim_{T\to\infty} \frac{1}{T}\int_{-T/2}^{T/2} x(t)x(t+\tau)\,\mathrm{d}t$$

$$= \lim_{T\to\infty} \frac{1}{T}\int_{-T/2}^{T/2} x(t)x(t+\tau+nT)\,\mathrm{d}t \tag{5.16}$$

$$r_{11}(\tau) = r_{11}(\tau + nT)$$

可以看出 $r_{11}(\tau)$是 τ 的周期函数，周期为 T。这是一个有用的性质，因为它使得小信噪比情况下噪声中周期信号的检测成为可能。波形自相关减少噪声，同时又得到了信号的周期自相关函数。一旦检测到信号，如果需要，进一步的处理可以用来确定信号的形状。

5.11 式说明了 $A\sin(\omega t)$的自相关函数是$(A^2/2)\cos(\omega\tau)$。在这种情况下，和别的情形下一样，自相关函数的幅度简单地与信号的幅度有关，因而可以用来估计信号的幅度。另一种常见的情况是幅度为 A 的方波的自相关函数是幅度为 A^2 的三角形波，读者可以自己证明。最后，要注意的是，自相关函数不是惟一的。这意味着许多不同的波形可能具有相同的自相关函数。因而从检测到的自相关函数想推导出波形的形状是不可能的。

现在考虑一种这样的情形：波形 $v(t)$是部分随机的。这代表着一个含噪声的信号，它可以写成一个信号项 $s(t)$和一个噪声项 $q(t)$之和。因此

$$v(t) = s(t) + q(t) \tag{5.17}$$

$s(t)$和 $q(t)$假定是不相关的。$v(t)$的抽样的自相关函数为 $r_{vv}(j)$，由下式给出，

$$r_{vv}(j) = \frac{1}{N}\sum_{n=0}^{N-1}[s(n)+q(n)][s(n+j)+q(n+j)] \tag{5.18}$$

$$= \frac{1}{N}\sum_{n=0}^{N-1} s(n)s(n+j) + \frac{1}{N}\sum_{n=0}^{N-1} s(n)q(n+j) + \frac{1}{N}\sum_{n=0}^{N-1} q(n)s(n+j) + \frac{1}{N}\sum_{n=0}^{N-1} q(n)q(n+j) \tag{5.19}$$

$$= r_{ss}(j) + E[s(n)q(n+j)] + E[q(n)s(n+j)] + E[q(n)q(n+j)]$$
$$= r_{ss}(j) + E[s(n)]E[q(n+j)] + E[q(n)]E[s(n+j)] + E[q(n)]E[q(n+j)]$$
$$= r_{ss}(j) + \overline{s(n)}\,\overline{q(n)} + \overline{q(n)}\,\overline{s(n)} + \overline{q(n)}^2 \tag{5.20}$$
$$= r_{ss}(j) + 2\bar{s}\bar{q} + \bar{q}^2$$

现在，对于大的N，$\bar{q}\to 0$，因此

$$r_{vv}(j) \to r_{ss}(j) \tag{5.21}$$

对于较小的N，5.19式中的互相关项和噪声的自相关将随延迟j的增大而趋向于零。

因此可以看出，一个部分随机的波形或者说含有噪声的波形的自相关函数是由信号部分的自相关函数加上一个噪声衰减函数组成。噪声衰减函数与随机和信号分量都有关，且衰减趋向于$2\bar{s}\bar{q} + \bar{q}^2$。因此，如果$|r_{ss}(j)| > |(2\bar{s}\bar{q} + \bar{q}^2)|$，$r_{vv}(j)$与$j$的关系图将显示出$s(t)$的周期性，如图5.11所示。这为我们提供了一种在噪声中识别一个信号的周期的方法（参见5.2.2节）。

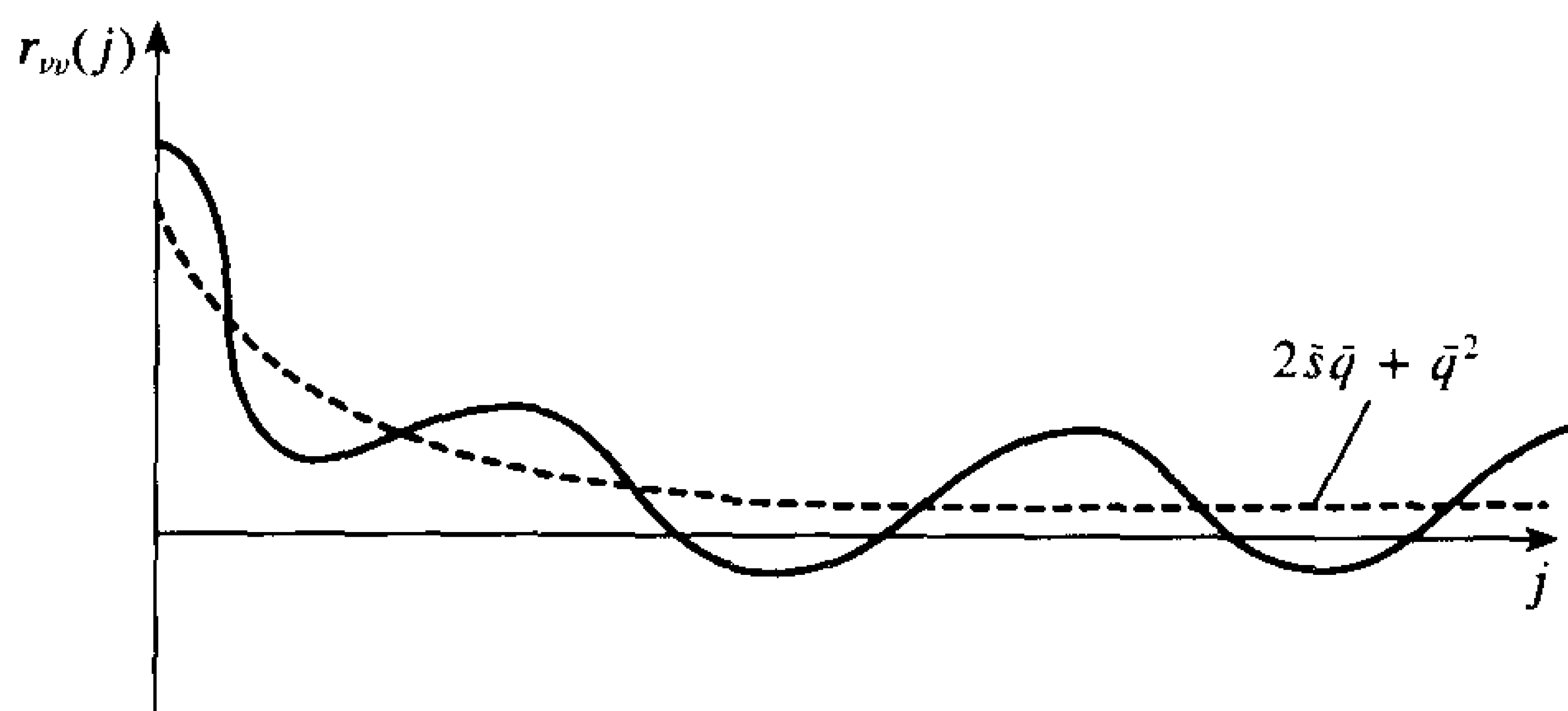


图5.11 一个含噪声信号的自相关函数

例5.4 推导两个含噪声波形的互相关函数。

令这两个波形分别是$\{s_1(t) + q_1(t)\}$和$\{s_2(t) + q_2(t)\}$。它们的抽样的互相关$r_{12}(j)$由下式给出：

$$r_{12}(j) = \frac{1}{N}\sum_{n=0}^{N-1}[\{s_1(n) + q_1(n)\}\{s_2(n+j) + q_2(n+j)\}] \tag{5.22}$$
$$= \frac{1}{N}\sum_{n=0}^{N-1}[s_1(n)s_2(n+j) + s_1(n)q_2(n+j) + q_1(n)s_2(n+j) + q_1(n)q_2(n+j)]$$
$$= \frac{1}{N}\sum_{n=0}^{N-1}s_1(n)s_2(n+j) + \frac{1}{N}\sum_{n=0}^{N-1}s_1(n)q_2(n+j) + \frac{1}{N}\sum_{n=0}^{N-1}q_1(n)s_2(n+j)$$
$$+ \frac{1}{N}\sum_{n=0}^{N-1}q_1(n)q_2(n+j)$$
$$= r_{s_1s_2}(j) + r_{s_1q_2}(j) + r_{q_1s_2}(j) + r_{q_1q_2}(j) \tag{5.23}$$

和前面的自相关的情形一样，5.23式的右边的后三项随着延时j的增大而趋向于零。对于大的N，5.23式变为

$$r_{12}(j) = r_{s_1s_2}(j) + \overline{s_1}\,\overline{q_2} + \overline{q_1}\,\overline{s_2} + \overline{q_1}\,\overline{q_2} \tag{5.24}$$

因而当N增加时，$r_{12}(j)\to r_{s_1s_2}(j)$，也就是趋向于两个信号的互相关。

上面的分析说明了互相关和自相关过程通过减少噪声分量而加强了信号特性。

5.2.2 相关的应用

5.2.2.1 能量谱密度和波形能量的计算

可以证明：

$$F[r_{11}(\tau)] = G_E(f) \tag{5.25}$$

其中 $G_E(f)$是波形的能量谱密度，即能量谱密度和自相关函数组成一个傅里叶变换对。

可以进一步证明：

$$r_{11}(0) = E \tag{5.26}$$

其中 E 是波形的总的能量。

例 5.5　求两个不同波形的零延时相关函数的关系式以及它们总能量的关系式。

令这两个波形是 $v_1(n)$和 $v_2(n)$，令它们的和是 $V(n) = v_1(n)+v_2(n)$。$V(n)$的零延迟的自相关函数是

$$r_{vv}(0) = E_V = \frac{1}{N}\sum_{n=0}^{N-1} V^2(n) = \frac{1}{N}\sum_{n=0}^{N-1} [v_1(n) + v_2(n)]^2$$

其中 E_V 是波形 $V(n)$的能量。

$$E_V = \frac{1}{N}\sum_{n=0}^{N-1} [v_1^2(n) + v_2^2(n) + 2v_1(n)v_2(n)]$$

$$= \frac{1}{N}\sum_{n=0}^{N-1} v_1^2(n) + \frac{1}{N}\sum_{n=0}^{N-1} v_2^2(n) + \frac{1}{N}\sum_{n=0}^{N-1} v_1(n)v_2(n)$$

所以，

$$E_V = r_{v_1}(0) + r_{v_2}(0) + 2r_{v_1v_2}(0) \tag{5.27}$$

5.27 式是要求的结果的第一种形式。另外它可以写成

$$E_V = E_{v_1} + E_{v_2} + 2r_{v_1v_2}(0) \tag{5.28}$$

因此，$V(n)$的能量等于它的分量的能量加上$2r_{v_1v_2}(n)$之和，其中 $r_{v_1v_2}(n)$是 $v_1(n)$和 $v_2(n)$的零延迟互相关函数。如果 $v_1(n)$和 $v_2(n)$是不相关的，那么整个能量恰好是各分量能量之和。

如果信号 $v_1(n)$和 $v_2(n)$是有噪声的，以至于 $v_1(n) = v_1'(n) + q_1(n)$ 以及 $v_2(n) = v_2'(n) + q_2(n)$，那么很容易证明：

$$E_V = E_{v_1'} + E_{v_2'} + E_{q_1} + E_{q_2} + r_{v_1'v_2'}(0) \tag{5.29}$$

5.2.2.2 噪声中周期信号的检测和估计

现在我们来考虑利用互相关来检测和估计噪声中的周期信号。首先提出的方法是这个掩藏在噪声中的信号能通过求它和一个可调整的“样板”（template）信号的互相关而估计出来。在先验知识的引导下，样板信号通过反复实验来调整，直到使互相关函数达到最大。这时，这个样板就是该信号的估计。参考 5.22 式，并且假定样板 $q_2(n) = 0$，这一方法就可以得到证明。那么，5.23 式变为

$$r_{12}(j) = r_{s_1s_2}(j) + r_{q_1s_2}(j) \tag{5.30}$$

$$= r_{s_1s_2}(j) + \bar{q}_1\bar{s}_2 \tag{5.31}$$

那么，当N增加时，$\bar{q}_1 \to 0$，

$$r_{12}(j) \to r_{s_1s_2}(j) \tag{5.32}$$

很显然，当$s_2(n)=s_1(n)$时，$r_{s_1s_2}(j)$是$s_1(n)$的自相关函数，$r_{s_1s_2}(j)$将达到最大。因此，改变样板$s_2(n)$的形状来使互相关函数最大，这样可以规定$s_2(n)$作为$s_1(n)$的估计。

信号估计的样板方法在有些时候是很方便的,例如当知道信号的形状近似为某种生物医学上的诱发电位时。但是我们喜欢选择一种更为科学的方法。在这种方法里，信号的周期首先通过对含噪声的波形的自相关运算而估计出来。接着对含噪声的波形和一个周期脉冲序列求互相关，这个脉冲序列的周期和信号的周期相等。最后得到的互相关函数就是信号的估计。

令$s(n)$是具有N_p个点（$N_p<N$）周期的信号，令噪声是$q(n)$，那么含噪声的波形是$S(n)=s(n)+q(n)$。令$\delta(n-kN_p)$是用来求互相关的周期脉冲序列，令N_δ是在互相关中用到的脉冲数目。这也等于含噪声的波形和脉冲序列互相关时信号的周期数。那么

$$r_{S\delta}(-j)=\frac{1}{N_\delta}\sum_{n=0}^{N-1}[s(n)+q(n)]\delta(n-kN_p-j),\quad k=0,1,2,\ldots \tag{5.33}$$

对于$j=0$，另外记着对于所有$n\neq kN_p$，有$\delta(n-kN_p)=0$，

$$\begin{aligned}r_{S\delta}(0)=&\frac{1}{N_\delta}[s(0)+q(0)+s(N_p)+q(N_p)+s(2N_p)+q(2N_p)+\ldots\\&+s(N)+q(N)]\end{aligned} \tag{5.34}$$

现在，因为信号$s(n+kN_p)=s(n)$的周期性，所以5.34式变为

$$r_{S\delta}(0)=\frac{1}{N_\delta}[Ns(0)+q(0)+q(N_p)+q(2N_p)+\ldots+q(N)]$$

或者

$$r_{S\delta}(0)=s(0)+\frac{1}{N_\delta}\sum_{k=0}^{N/N_p}q(kN_p) \tag{5.35}$$

当$N\to\infty$，$(1/N_\delta)\sum_{k=0}^{N/N_p}q(kN_p)\to 0$，因此$r_{S\delta}(0)\to s(0)$。类似地，对于其他$j$值，

$$r_{S\delta}(-j)=\frac{1}{N_\delta}\sum_{n=0}^{N-1}[s(n)+q(n)]\delta[(n-j)-kN_p],\quad k=0,1,2,\ldots$$

这同样导致了噪声的对消，得到了$s(n)$（$n=1,2,\ldots$）的值。因而，由5.33式，

$$r_{S\delta}(-j)=s(0),s(1),\ldots,s(N-1),\quad j=0,1,2,\ldots$$

这就是要求的信号。因而，一个在噪声波形中丢失的信号可以通过如下步骤来估计：

(1) 对波形求自相关，求出信号的周期；

(2) 用一个和信号等周期的脉冲串来和波形求互相关。在这个过程中，脉冲串相对于信号波形向右平移。

5.2.2.3 匹配滤波器在相关检测中的实现

相关的另外一种应用是匹配滤波器的相关检测实现。匹配滤波器是使输出端S/N最大的滤波器。匹配滤波器的冲激响应为（Stremler, 1982）

$$h(t)=cs_1(T-t) \tag{5.36}$$

其中 c 是一个任意常数，$s_i(t)$是输入（无噪声）信号，由下式给出：

$$s_i(t) = s_1(t) \quad 对于\ 0 \leqslant t \leqslant T$$
$$= 0 \quad 对于\ T < t < 0$$

T是滤波器的输出被抽样的时刻。首先在时间上反转信号，接着把它沿着时间轴推进 T(s)，可以看出这样我们就得到了冲激响应。例如，图 5.12(a)给出了一个信号，它实际上是一个 8 位的 PCM 码字，图 5.12(b)给出了使这个信号的检测达到最大的匹配滤波器的响应。

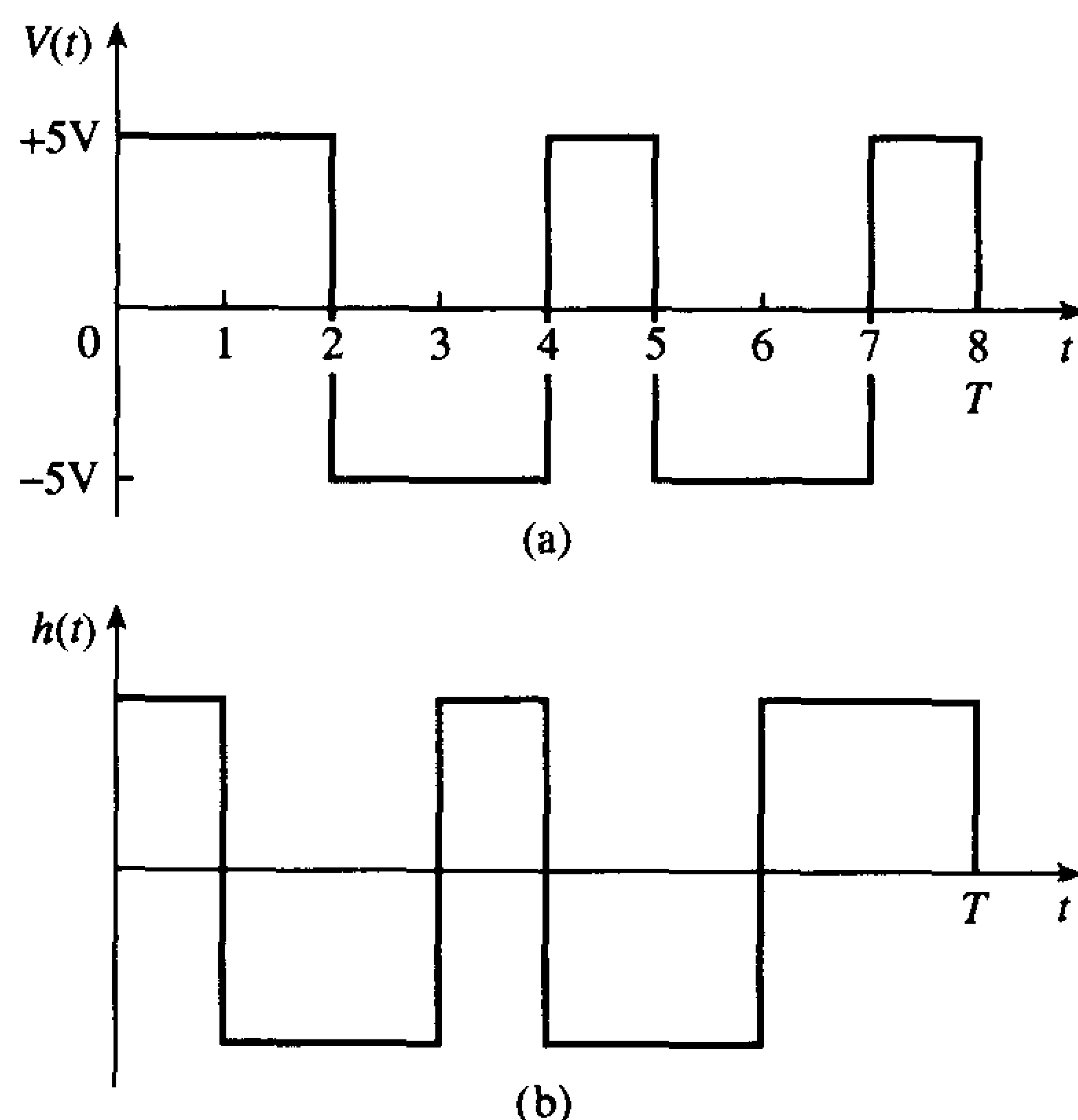


图 5.12　(a)8 位码字的信号；(b)对应的匹配滤波器的冲激响应

现在可以看出匹配滤波器检测等价于相关。滤波器的输出 $y(t)$首先可以表示为输入 $s(t)$和它的冲激响应的卷积（卷积参见第 5.3 节）：

$$y(t) = \int_{-\infty}^{\infty} s(\tau)h(t-\tau)\,\mathrm{d}\tau \tag{5.37}$$

其中

$$s(t) = s_1(t) + q(t) \tag{5.38}$$

τ是延迟，如通常的那样，$q(t)$代表噪声分量。把 5.38 式代入到 5.37 式中，得

$$y(t) = \int_{-\infty}^{\infty} [s_1(\tau) + q(\tau)]h(t-\tau)\,\mathrm{d}\tau$$
$$= \int_{-\infty}^{\infty} s_1(\tau)h(t-\tau)\,\mathrm{d}\tau + \int_{-\infty}^{\infty} q(\tau)h(t-\tau)\,\mathrm{d}\tau$$

因为 $q(\tau)$是随机的，所以上式右端的第二项趋向于零，并且它和 $h(t-\tau)$是不相关的。因此

$$y(t) \approx \int_{-\infty}^{\infty} s_1(\tau)h(t-\tau)\,\mathrm{d}\tau \tag{5.39}$$

现在，由 5.36 式，

$$h(t-\tau) = cs_1(T-t+\tau) \tag{5.40}$$

把5.39式和5.40式合并，得

$$y(t) \approx \int_{-\infty}^{\infty} s_1(\tau) c s_1(T-t+\tau)\,\mathrm{d}\tau \tag{5.41}$$

如果这个输出在$t=T$时刻抽样，且$c=1$，那么

$$\begin{aligned} y(T) &\approx \int_{-\infty}^{\infty} s_1(\tau) c s_1(\tau)\,\mathrm{d}\tau \\ &\approx \int_{-\infty}^{\infty} s_1^2(\tau)\,\mathrm{d}\tau = \int_{-\infty}^{\infty} s_1^2(t)\,\mathrm{d}t = r_{11}(0) \end{aligned} \tag{5.42}$$

因此，$y(T)$是$s_1(t)$的零延时自相关，它可以通过对含噪声的输入和本地产生的不含噪声的信号求互相关而得到，这构成了相关检测器。图5.13给出了一个相关检测器的框图。例如，一个PCM码字检测器对每一个码字都将包含一个相关检测器，如图5.14所示。

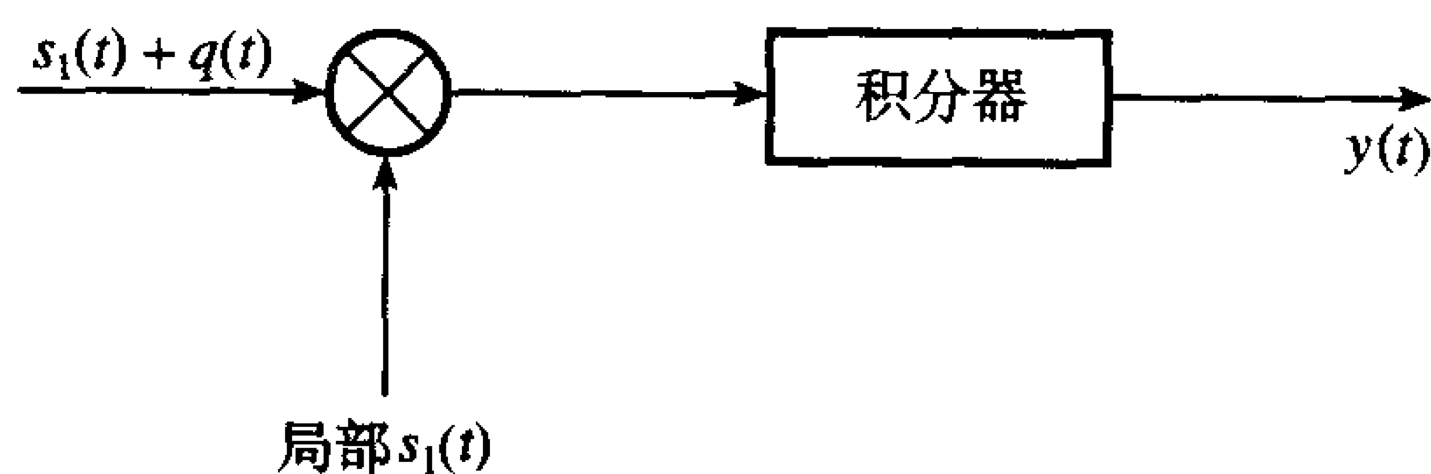


图5.13　相关检测器的框图

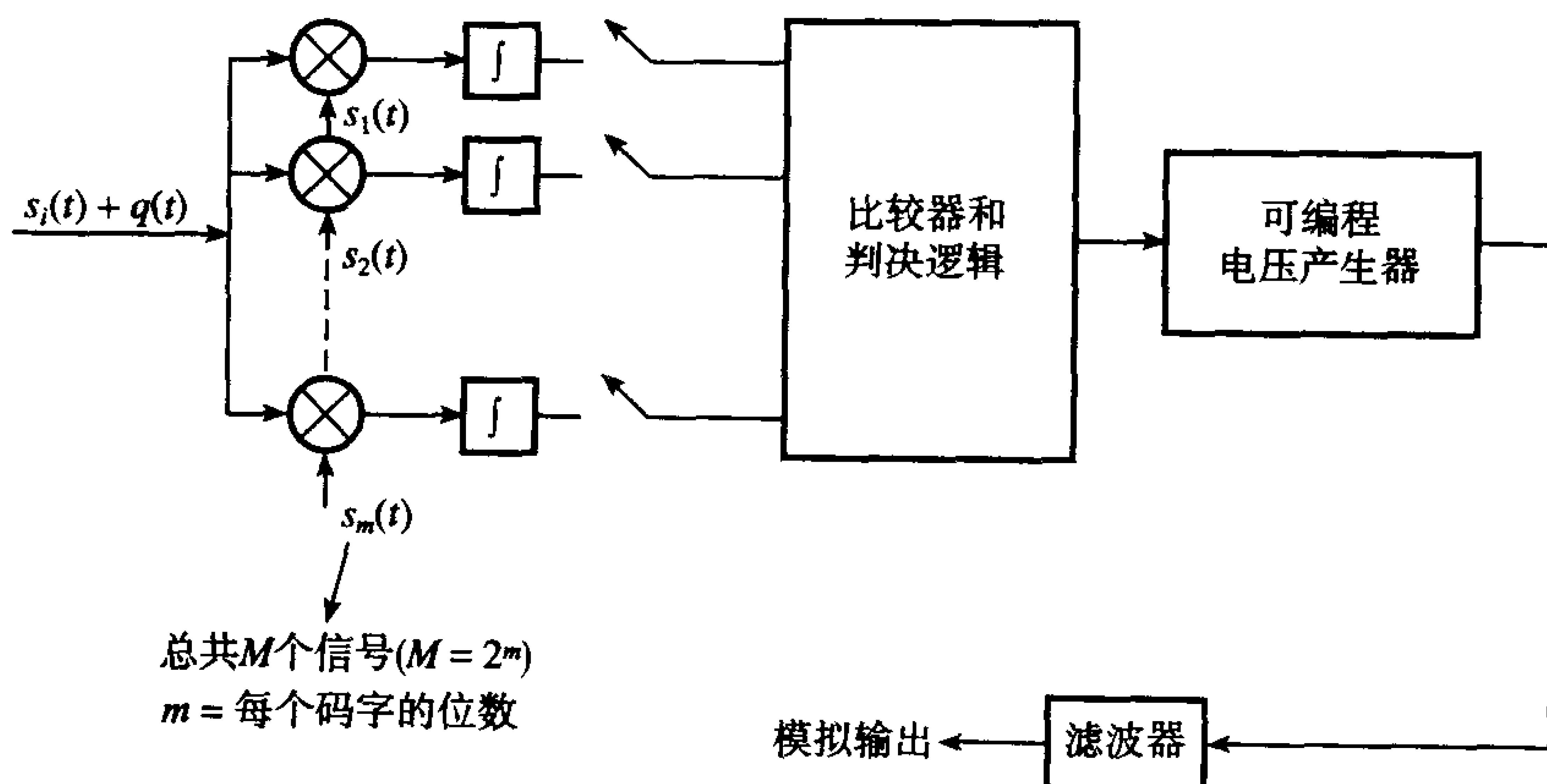


图5.14　一个基于相关检测器的PCM码字检测器

在一个数字式m位的码字检测器中，码字被储存，然后与输入的位数m相乘。只要出现下面的情况，就会出现峰值：(i) m个输入位严格地对应于m位码字，(ii) m个输入位碰巧对应于m位码字。第二种情况是极不希望的。这种情况出现在如果两个相邻的码字碰巧包含一个和要求的m位码字相同的位序列，或者如果一个码字受到污染。这种可能性使得我们在相关接收机中必须既要安排字同步，又要安排位同步。

这导致了对同步码字的要求，这些同步码字具有如下的性质：

(1) 当抽样时间$t \neq T$，相关很小；

(2) 当抽样时间取$t=T$，相关比较大。

具有这些性质的码字将在零延时有大的自相关值，但在其他的延迟的自相关较小。因此，在接收机检测到一个大的互相关值意味着输入的码字和储存的码字对齐了，这就使接收机达到同步。随机波形具有这种自相关性质，它可以在数字接收机中用伪噪声（PN）序列实现，伪噪声序列很容易从一个多抽头移位寄存器产生。图 5.15 给出一个三阶 PN 序列产生器的例子。产生的输出序列是 1, 1, 1, 0, 0, 1, 0，这个序列是重复的。当这个序列表示为双极波形时就产生了自相关函数，如图 5.16 所示。

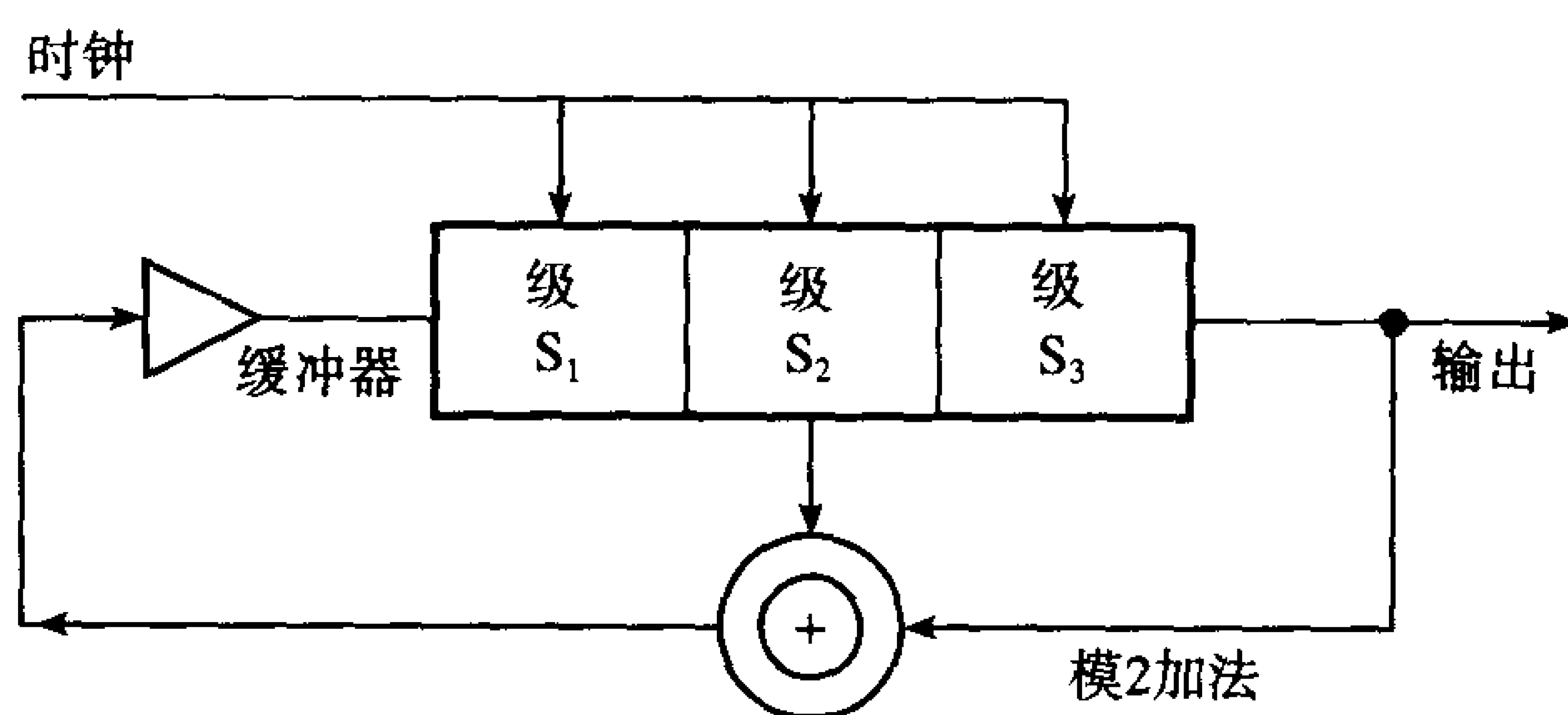


图 5.15　一个三阶伪噪声序列产生器

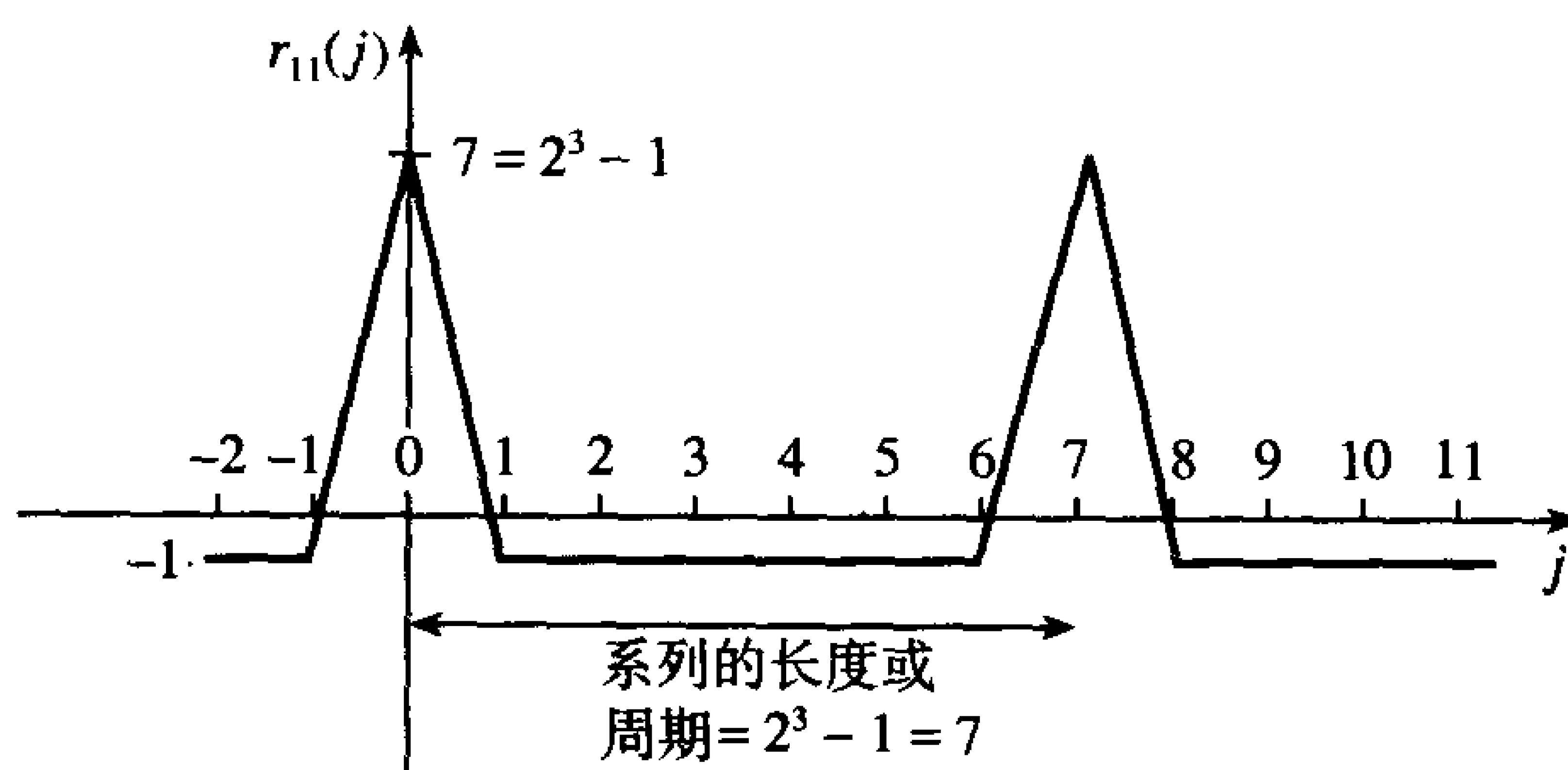


图 5.16　一个双极波形三级伪噪声产生器的自相关函数

PN 序列的一些性质如下：

(1) m 位码字产生一个长度为 2^m-1 的序列。

(2) 峰值是 2^m-1。

(3) 峰值之外的其他自相关函数等于 -1。

(4) 输出序列包含 2^{m-1} 个 1 和 $2^{m-1}-1$ 个零。

(5) 它们的功率谱密度是均匀的，所以它们可以用做白噪声源。

最后一条性质提供了 PN 序列的另外一个应用：可以作为白噪声源。

5.2.2.4　电子系统冲激响应的确定

相关和PN序列可进一步应用到电子系统冲激响应的确定中。但这在冲激测试系统里可能有些困难。例如，在有噪声的情况下，小的冲激可能被噪声遮挡，而大的冲激又有可能引起系统的过载。利用单个冲激在整个带宽里保持均匀的能量谱密度也是有困难的。然而，PN 序列如前面解释过的那样，具有均匀的能量谱。另外，如果测量时间是序列长度的倍数，则测量中的方差将是零，这导致短的测量时间和高的测量精度。

这种方法的原理是把PN序列作为系统的输入。那么冲激响应将由提供的序列和输出的序列互相关给出。证明过程如下。

令 $q(t)$为输入的 PN 序列，并且令 $y(t)$为系统的输出，系统的冲激响应为 $h(t)$，那么

$$r_{qy}(\tau) = \lim_{T\to\infty} \frac{1}{T}\int_0^T q(t)y(t+\tau)\,\mathrm{d}t \tag{5.43}$$

$$= \lim_{T\to\infty} \frac{1}{T}\int_0^T q(t)\,\mathrm{d}t\int_{-\infty}^{\infty} h(v)q(t-v+\tau)\,\mathrm{d}v \tag{5.44}$$

其中的 $y(t)$是由输入和冲激响应的卷积给出的：

$$y(t) = \int_{-\infty}^{\infty} h(v)q(t-v)\,\mathrm{d}v \tag{5.45}$$

在 5.44 式中，改变积分次序得

$$r_{qy}(\tau) = \int_{-\infty}^{\infty} h(v)\,\mathrm{d}v \lim_{T\to\infty} \frac{1}{T}\int_0^T q(t)q(t-v+\tau)\,\mathrm{d}t \tag{5.46}$$

$$= \int_{-\infty}^{\infty} h(v)r_{qq}(\tau-v)\,\mathrm{d}v \tag{5.47}$$

$r_{qq}(\tau\text{-}v)$近似为 δ 函数，因为它是 PN 序列的自相关函数。因此，5.47 式可以表示为

$$r_{qy}(t) = K\int_{-\infty}^{\infty} h(v)\delta(\tau-v)\,\mathrm{d}v = Kh(t) \tag{5.48}$$

其中 K 是冲激函数的面积，且等于噪声的均方根值（Beauchamp, 1973）。图 5.17 解释了这种电路。这种方法可能会引起一些误差，应该采用预防措施来避免这些误差。

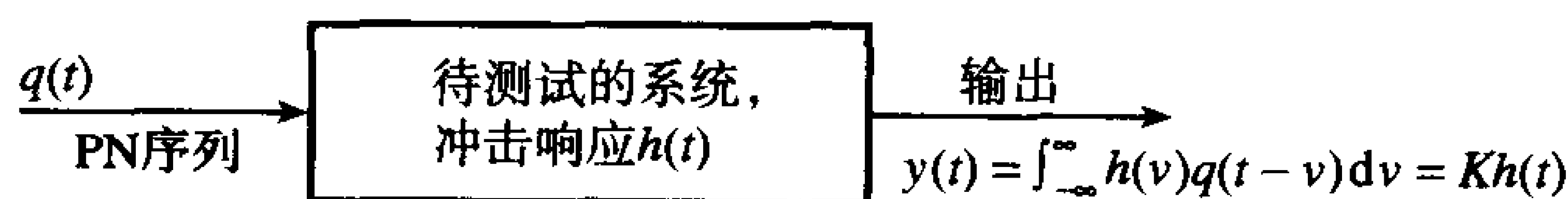


图 5.17　一个电子系统的冲激响应的确定

5.2.2.5　含噪声的周期信号信噪比的确定

通过测量含噪声的周期信号的相关系数，可以确定它的信噪比以及信号功率、噪声功率（Main and Howell, 1993）。这些表达式的推导如下所示。我们将会发现含噪声的周期信号正如5.2.2.2节里描述的那样。

令 $V_s(i)$表示信号，$V_n(i)$表示噪声，那么含噪声的信号 $V(i)$为

$$V(i) = V_s(i) + V_n(i) \tag{5.49}$$

那么，对周期为 n 个抽样间隔的周期信号，

$$V(i) = V(i+n) \tag{5.50}$$

现在 $V(i)$的方差可以定义为

$$\mathrm{Cov}\,[V(i)] = \mathrm{Cov}\,[V(i),\ V(i+n)] = \frac{1}{N}\sum_{i=1}^{N}[\{V(i)-\bar{V}(i)\}\{V(i+n)-\bar{V}(i+n)\}] \tag{5.51}$$

其中

$$\bar{V}(i) = \frac{1}{N}\sum_{i=1}^{N} V(i)$$

是 $V(i)$的均值。因此，$\mathrm{Cov}[V(i)]$可以看作为 $V(i)$的 n 延时的零均值自相关函数。那么 $V(i)$的自相关系数可以按通常的方式定义为

$$\rho[V(i)]=\frac{\mathrm{Cov}[V(i)]}{\sqrt{\left\{\frac{1}{N}\sum_{i=1}^{N}(V(i)-\bar{V}(i))^2\right\}\left\{\frac{1}{N}\sum_{i=1}^{N}(V(i+n)-\bar{V}(i+n))^2\right\}}}=\frac{\mathrm{Cov}[V(i)]}{\sigma[V(i)]\sigma[V(i+n)]} \tag{5.52}$$

其中

$$\sigma[V(i)]=\sqrt{\frac{1}{N}\sum_{i=1}^{N}(V(i)-\bar{V}(i))^2} \tag{5.53}$$

是 $V(i)$的标准差。

注意在5.51式到5.53式中，通过因子 N 归一化给出的是有偏估计。用 $N-1$ 代替 N 可以得到无偏的估计。我们也注意到类似于下式的项：

$$\sigma^2[V_x(i)]=\frac{1}{N}\sum_{i=1}^{N}\left(V_x(i)-\bar{V}_x(i)\right)^2=\mathrm{Var}[V_x(i)]$$

称为方差，它表示和 $V_x(i)$有关的功率。

随着抽样数目的增加，含噪声信号之间的相关趋向于零。在此条件下展开5.51式，因此有

$$\begin{aligned}\mathrm{Cov}[V(i),V(i+n)]&=\mathrm{Cov}[(V_s(i)+V_n(i)),(V_s(i+n)+V_n(i+n))]\\&=\mathrm{Cov}[V_s(i),V_s(i+n)]=\mathrm{Var}[V_s(i)]\end{aligned} \tag{5.54}$$

对于足够长的数据长度，$\sigma[V_n(i)]=\sigma[V_n(i+n)]$，由于 $\mathrm{Cov}[V_s(i),V_n(i)]=0$，所以

$$\begin{aligned}\sigma[V(i)]\sigma[V(i+n)]&=\sigma^2[V(i)]=\mathrm{Var}[V(i)]=\mathrm{Var}[V_s(i)+V_n(i)]\\&=\mathrm{Var}[V_s(i)]+\mathrm{Var}[V_n(i)]+2\mathrm{Cov}[V_s(i),V_n(i)]=\mathrm{Var}[V_s(i)]+\mathrm{Var}[V_n(i)]\end{aligned} \tag{5.55}$$

通过把5.54式和5.55式代入到5.52式，可以得到自相关系数的表达式：

$$\rho[V(i)]=\frac{\mathrm{Var}[V_s(i)]}{\mathrm{Var}[V_s(i)]+\mathrm{Var}[V_n(i)]}=\frac{1}{1+\dfrac{\mathrm{Var}[V_n(i)]}{\mathrm{Var}[V_s(i)]}} \tag{5.56}$$

信噪比 S/N (dB)定义为 $10\log_{10}\{(\text{信号功率})/(\text{噪声功率})\}$ dB，用现在的符号表示为

$$\frac{S}{N}(\mathrm{dB})=10\log_{10}\left|\frac{\mathrm{Var}[V_s(i)]}{\mathrm{Var}[V_n(i)]}\right|\mathrm{dB} \tag{5.57}$$

合并并且替换5.56式和5.57式，得

$$\frac{S}{N}(\mathrm{dB})=10\log_{10}\left|\frac{\rho[V(i)]}{1-\rho[V(i)]}\right|\mathrm{dB} \tag{5.58}$$

因此，含噪声的周期波形的信噪比就可以很容易地从它的自相关系数得到。

由5.55式，我们有

$$\mathrm{Var}[V_s(i)]+\mathrm{Var}[V_n(i)]\equiv S+N=\mathrm{Var}[V(i)] \tag{5.59}$$

其中 S 和 N 代表信号和噪声的功率。利用5.56式和5.59式可以推导出 S 和 N，

$$S=\rho[V(i)]\ \mathrm{Var}[V(i)] \tag{5.60}$$

和

$$N=(1-\rho[V(i)])\ \mathrm{Var}[V(i)] \tag{5.61}$$

5.58 式已经应用到根据信噪比来评估磁记录通道的性能（Main and Howell, 1993）。

5.2.3 快速相关

利用相关定理，可以使相关计算加速，通常可表示为

$$r_{12}(j)=F_{\mathrm{D}}^{-1}[X_1^*(k)X_2(k)] \tag{5.62}$$

然而确切的应该写成

$$r_{12}(j)=\frac{1}{N}F_{\mathrm{D}}^{-1}[X_1^*(k)X_2(k)] \tag{5.63}$$

其中 F_{D}^{-1} 表示离散傅里叶反变换。这种方法要求两个离散傅里叶变换（DFT）和一个 DFT 反变换，它们的每一个都非常易于用一个 FFT 方法（参见第 3 章）来计算。如果在这个序列里项的数目足够大，利用这种 FFT 方法比直接用互相关求要更快。

相关定理的证明

令 $x_1(l)$、$x_2(r)$ 和 $x_3(n)$ 是长度为 N 的周期序列，令它们的 DFT 分别是 $X_1(k)$、$X_2(k)$ 和 $X_3(k)$。另外，令

$$X_3(k)=X_1^*(k)X_2(k) \tag{5.64}$$

而

$$X_1^*(k)=\sum_{l=0}^{N-1}x_1(l)\,\mathrm{e}^{\mathrm{j}(2\pi/N)lk} \tag{5.65}$$

和

$$X_2(k)=\sum_{r=0}^{N-1}x_2(r)\,\mathrm{e}^{\mathrm{j}(2\pi/N)(-rk)} \tag{5.66}$$

将 5.56 式和 5.66 式代入到 5.64 式，得

$$X_3(k)=\sum_{l=0}^{N-1}x_1(l)\,\mathrm{e}^{\mathrm{j}(2\pi/N)lk}\sum_{r=0}^{N-1}x_2(r)\,\mathrm{e}^{\mathrm{j}(2\pi/N)(-rk)} \tag{5.67}$$

$$=\sum_{l=0}^{N-1}\sum_{r=0}^{N-1}x_1(l)x_2(r)\,\mathrm{e}^{\mathrm{j}(2\pi/N)(lk-rk)} \tag{5.68}$$

现在

$$x_3(n)=\frac{1}{N}\sum_{k=0}^{N-1}X_3(k)\,\mathrm{e}^{\mathrm{j}(2\pi/N)nk} \tag{5.69}$$

所以，将 5.68 式代入到 5.69 式中，得

$$x_3(n)=\frac{1}{N}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1}\sum_{r=0}^{N-1}x_1(l)x_2(r)\,\mathrm{e}^{\mathrm{j}(2\pi/N)(lk-rk+nk)}$$

$$=\frac{1}{N}\sum_{l=0}^{N-1}x_1(l)\sum_{r=0}^{N-1}x_2(r)\left[\sum_{k=0}^{N-1}\mathrm{e}^{\mathrm{j}(2\pi/N)(l-r+n)k}\right] \tag{5.70}$$

当 $r=n+l$ 时，方括号中的项等于 N。当 $r \neq n+l$ 时，将其看成一个具有下面形式的几何级数，

$$\sum ax^n$$

上式 N 项加起来，得

$$\frac{a(1-x^N)}{1-x}$$

在这种情况下，和变为

$$\frac{1[1-\mathrm{e}^{\mathrm{j}(2\pi/N)(l-r+n)N}]}{1-\mathrm{e}^{\mathrm{j}(2\pi/N)(l-r+n)}} \tag{5.71}$$

分子里的指数通常是 2π 的整数倍，所以指数项为 1。因此，当 $r \neq n+l$ 时，和值等于零。5.70式可以写成

$$x_3(n)=\frac{1}{N}\sum_{l=0}^{N-1}x_1(l)\sum_{r=0}^{N-1}x_2(r)N\delta(l-r+n) \tag{5.72}$$

在这个式子里，当 $r=n+l$ 时，$\delta(l-r+n)=1$；当 $r \neq n+l$，$\delta(l-r+n)=0$。化简并把 $r=n+l$ 代入，得

$$x_3(n)=\sum_{l=0}^{N-1}x_1(l)x_2(l+n) \tag{5.73}$$

或者

$$\frac{1}{N}x_3(n)=\frac{1}{N}\sum_{l=0}^{N-1}x_1(l)x_2(l+n) \tag{5.74}$$

这个等式的右边等价于 $x_1(n)$ 和 $x_2(n)$ 的互相关，可以看出它等于 $(1/N)\,x_3(n)$。由 5.69 式，

$$x_3(n)=F_{\mathrm{D}}^{-1}[X_3(k)] \tag{5.75}$$

因此，通过合并 5.74 式、5.75 式和 5.63 式，

$$\frac{1}{N}F_{\mathrm{D}}^{-1}[X_3(k)]=r_{12}(n)=\frac{1}{N}F_{\mathrm{D}}^{-1}[X_1^*(k)X_2(k)] \tag{5.76}$$

最后，用 j 代替 n，得

$$r_{12}(j)=\frac{1}{N}F_{\mathrm{D}}^{-1}[X_1^*(k)X_2(k)] \tag{5.77}$$

例 5.6　应用相关定理求下列两个序列 $x_1(n)$ 和 $x_2(n)$ 的互相关：

$$x_1(n)=\{1,0,0,1\}$$

$$x_2(n)=\{0.5,1,1,0.5\}$$

首先利用相关定理 5.77 式，在 3.5 节里的 $X_1(k)$ 已求得为

$$X_1(k)=2,\ 1+\mathrm{j},\ 0,\ 1-\mathrm{j}$$

所以，

$$X_1^*(k)=2,\ 1-\mathrm{j},\ 0,\ 1+\mathrm{j}$$

利用第 3.5 节给出的 FFT 算法，非常容易求得 $X_2(k)$。因此，当 $x_0=0.5$、$x_2=1$、$x_1=1$ 和 $x_3=0.5$ 时，

$$X_{21}(0) = x_0 + x_2 = 1.5$$
$$X_{21}(1) = x_0 - x_2 = -0.5$$
$$X_{22}(0) = x_1 + x_3 = 1.5$$
$$X_{22}(1) = x_1 - x_3 = 0.5$$
$$X_{11}(0) = X_{21}(0) + X_{22}(0) = 3$$
$$X_{11}(1) = X_{21}(1) + (-\mathrm{j})X_{22}(1) = -0.5 - \mathrm{j}0.5$$
$$X_{11}(2) = X_{21}(0) - X_{22}(0) = 0$$
$$X_{11}(3) = X_{21}(1) - (-\mathrm{j})X_{22}(1) = -0.5 + \mathrm{j}0.5$$

把 FFT 的值放在一起，得

$$X_1^*(k) = 2, 1 - \mathrm{j}, 0, 1 + \mathrm{j}$$
$$X_2(k) = 3, -0.5 - \mathrm{j}0.5, 0, -0.5 + \mathrm{j}0.5$$

所以

$$X_1^*(0)X_2(0) = 2 \times 3 = 6$$
$$X_1^*(1)X_2(1) = (1 - \mathrm{j})(-0.5 - \mathrm{j}0.5) = -1$$
$$X_1^*(2)X_2(2) = 0 \times 0 = 0$$
$$X_1^*(3)X_2(3) = 0.5(1 + \mathrm{j})(-1 + \mathrm{j}) = -1$$

因此

$$[X_1^*(k)X_2(k)] = 6, -1, 0, -1$$

现在有必要对它取DFT反变换（IDFT）。如第 3.6 节解释的那样，通过改变上面FFT算法里指数项（在加权因数 W_N 里）的符号，并用 N 去除结果。因此，不必在算法里改变标号，

$$X_{21}(0) = x_0 + x_2 = 6$$
$$X_{21}(1) = x_0 - x_2 = 6$$
$$X_{22}(0) = x_1 + x_3 = -2$$
$$X_{22}(1) = x_1 - x_3 = 0$$
$$X_{11}(0) = X_{21}(0) + X_{22}(0) = 4$$
$$X_{11}(1) = X_{21}(1) + \mathrm{j}X_{22}(1) = 6$$
$$X_{11}(2) = X_{21}(0) - X_{22}(0) = 8$$
$$X_{11}(3) = X_{21}(1) - \mathrm{j}X_{22}(1) = 6$$

通过把值 $X_{11}(0)$、$X_{11}(1)$、$X_{11}(2)$和 $X_{11}(3)$除以 $N = 4$，得到 $F_D^{-1}[X_1^*(k)X_2(k)]$ 的分量。因此

$$F_D^{-1}[X_1^*(k)X_2(k)] = 1, 1.5, 2, 1.5$$

由 5.77 式，

$$r_{12}(j) = \frac{1}{4}F_D^{-1}[X_1^*(k)X_2(k)] = \{0.25, 0.375, 0.5, 0.375\} \tag{5.78}$$

这种相关将是循环的，因为所有的数据都是以 N 为周期的。互相关 $r_{12}(j)$可以直接计算出为

$$r_{12}(0)=(1\times 0.5+0+0+1\times 0.5)/4=0.25$$
$$r_{12}(1)=(1\times 1+0+0+1\times 0.5)/4=0.375$$
$$r_{12}(2)=(1\times 1+0+0+1\times 1)/4=0.5$$
$$r_{12}(3)=(1\times 0.5+0+0+1\times 1)/4=0.375$$

下一个值$r_{12}(4)$是0.25，和$r_{12}(0)$一样，序列周期性地重复。这就是5.2.1节里讨论过的循环相关，而且这个结果和上面利用相关定理推导出的结果是一致的。如同5.2.1节解释的那样，通过对两个序列增加零，可以利用相关定理来获得线性相关。因此，如果序列$x_1(n)$长度是N_1，$x_2(n)$的长度是N_2，那么对$x_1(n)$增加N_2-1个零，对$x_2(n)$增加N_1-1个零。接着利用这两个增加了零的序列计算互相关。这种利用相关定理和FFT求互相关的方法称之为快速相关。

也可以通过递归实现来加速互相关计算，对零延时的情况我们将做出解释。在零延时两个抽样波形$x_1(n)$和$x_2(n)$的互相关为

$$r_{12}(0)=\frac{1}{N}\sum_{n=0}^{N-1}x_1(n)x_2(n) \tag{5.79}$$

这包括N个乘积计算、$N-1$个加法计算和一个除法。当用于在线应用时，新的数字以抽样速率到达，那么这样计算将占用过多的时间。当下一个数据对可用时，计算必须重复。新的计算和前面的计算惟一不同的地方在于：新的数据的乘积要加到乘积对的和中，并且必须减去第一个乘积。因此，对每一个互相关：

$$\text{新值}=\text{以前的值}+\frac{1}{N}(\text{两个新的数据的乘积})-\frac{1}{N}(\text{最早的两个数据的乘积}) \tag{5.80}$$

这是递归算法的基础。假如保存了数据对的乘积，那么每一个互相关仅需要一个乘法、一个减法、一个加法和一个除法。对于N点相关，在前$N-1$个点被计算出来后，递归方法就可以给出正确的值。

在许多应用中要求令数据的均值为零，例如从电子波形中移去dc电平。需要计算波形的均值，接着从所有的抽样值中减去它。这个均值计算也可以递归地计算，因为对每一个新的数据对有

$$\text{新均值}=\text{以前的均值}+\frac{1}{N}(\text{新数据}-\text{第一个数据}) \tag{5.81}$$

也有可能把减去均值电平和互相关计算合并在一个递归算法里。考虑

$$\bar{x}_1(k)=\frac{1}{N}\sum_{n=0}^{N-1}x_1(n) \tag{5.82}$$

和

$$\bar{x}_2(k)=\frac{1}{N}\sum_{n=0}^{N-1}x_2(n) \tag{5.83}$$

N点第k个序列的互相关函数的值为

$$r_{12}(k)=\frac{1}{N}\sum_{n=0}^{N-1}x_1(n)x_2(n) \tag{5.84}$$

当均值被移出以后，互相关函数的值变为$r_{12}^0(k)$，

$$r_{12}^0(k)=\frac{1}{N}\sum_{n=0}^{N-1}[x_1(n)-\bar{x}_1(k)][x_2(n)-\bar{x}_2(k)] \tag{5.85}$$

展开并化简，得

$$r_{12}^{0}(k) = r_{12}(k) - \bar{x}_1(k)\bar{x}_2(k) \tag{5.86}$$

合并 5.80 式和 5.83 式，有

$$r_{12}(k) = r_{12}(k-1) + \frac{1}{N}[x_1(k)x_2(k) - x_1(k-N)x_2(k-N)] \tag{5.87}$$

由 5.81 式，

$$\bar{x}_1(k) = \bar{x}_1(k-1) + \frac{1}{N}[x_1(k) - x_1(k-N)] \tag{5.88}$$

和

$$\bar{x}_2(k) = \bar{x}_2(k-1) + \frac{1}{N}[x_2(k) - x_2(k-N)] \tag{5.89}$$

5.86式~5.89式组成了递归算法，该算法将从数据中减去均值和互相关的计算合并在一起。每一次计算只要求三次乘法、四次减法、三次加法和四次除法。当数据的均值是变化着的时，N的选择应该引起注意，否则可能会得到错误的结果。

5.3 卷积描述

卷积这个词是描述系统的输入如何与系统相互作用产生输出。通常来说，系统的输出将是输入的延迟、衰减或者放大。当系统的输入是冲激信号的时候，考虑系统的输出在实践中是非常有用的。这是因为任何输入可以用一系列不同强度的冲激信号来表示。当系统的输入为冲激信号时，系统的输出将不是对应的冲激信号，而是随时间变化，经过一个最大值，如图5.18所示。这个图说明了在抽样时刻m时刻提供的输入是单位冲激，在抽样时刻m，得到的输出是$h(m)$。这个特性称为系统的冲激响应$h(m)$。

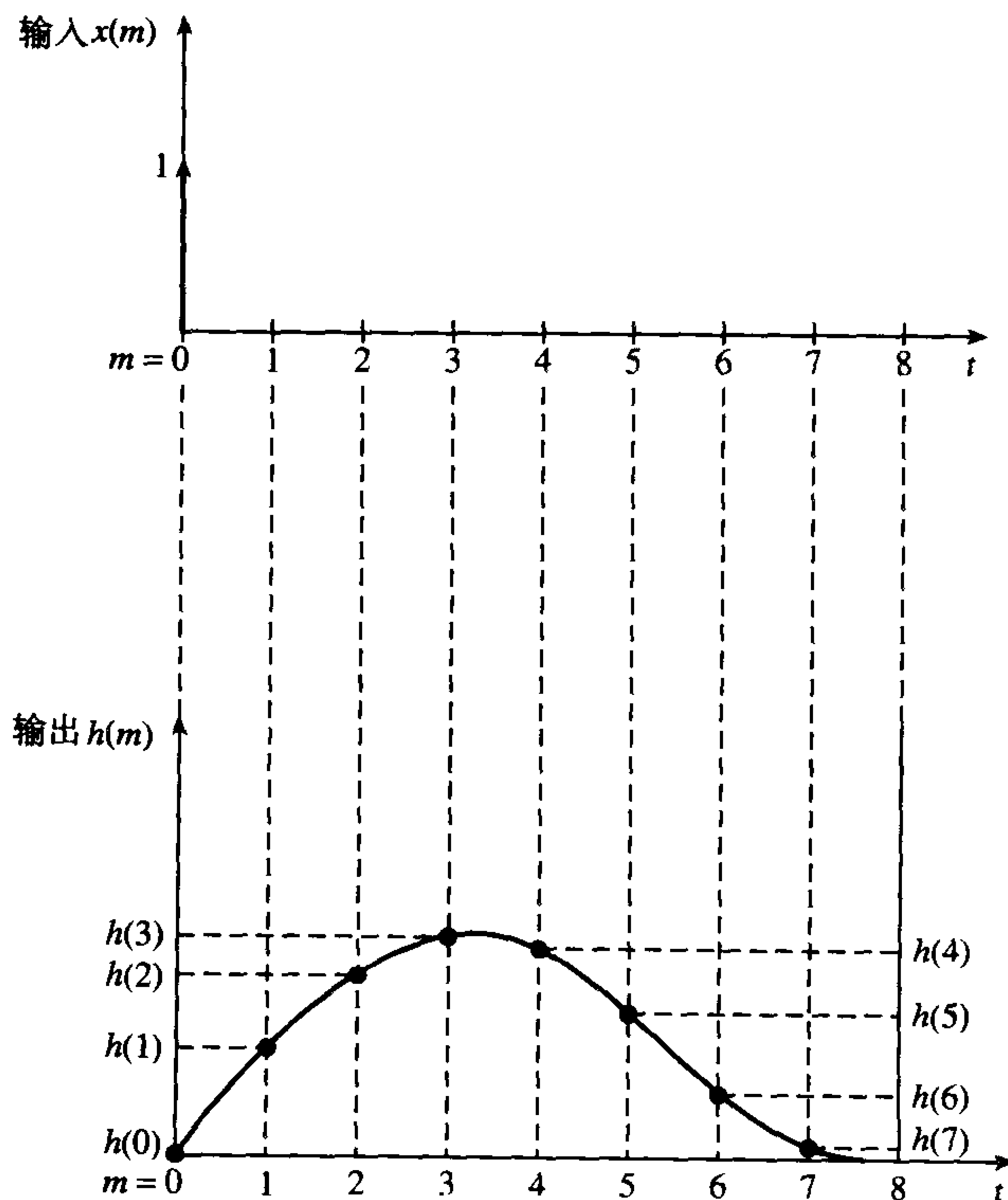


图 5.18 冲激输入和系统相应的冲激响应

现在考虑把一个冲激序列 $x(m)$ 加到系统，在抽样瞬间 m 加入。参考图5.19，在0瞬间的输出是 $y(0)$，它由下式给定：

$$y(0) = h(0)x(0)$$

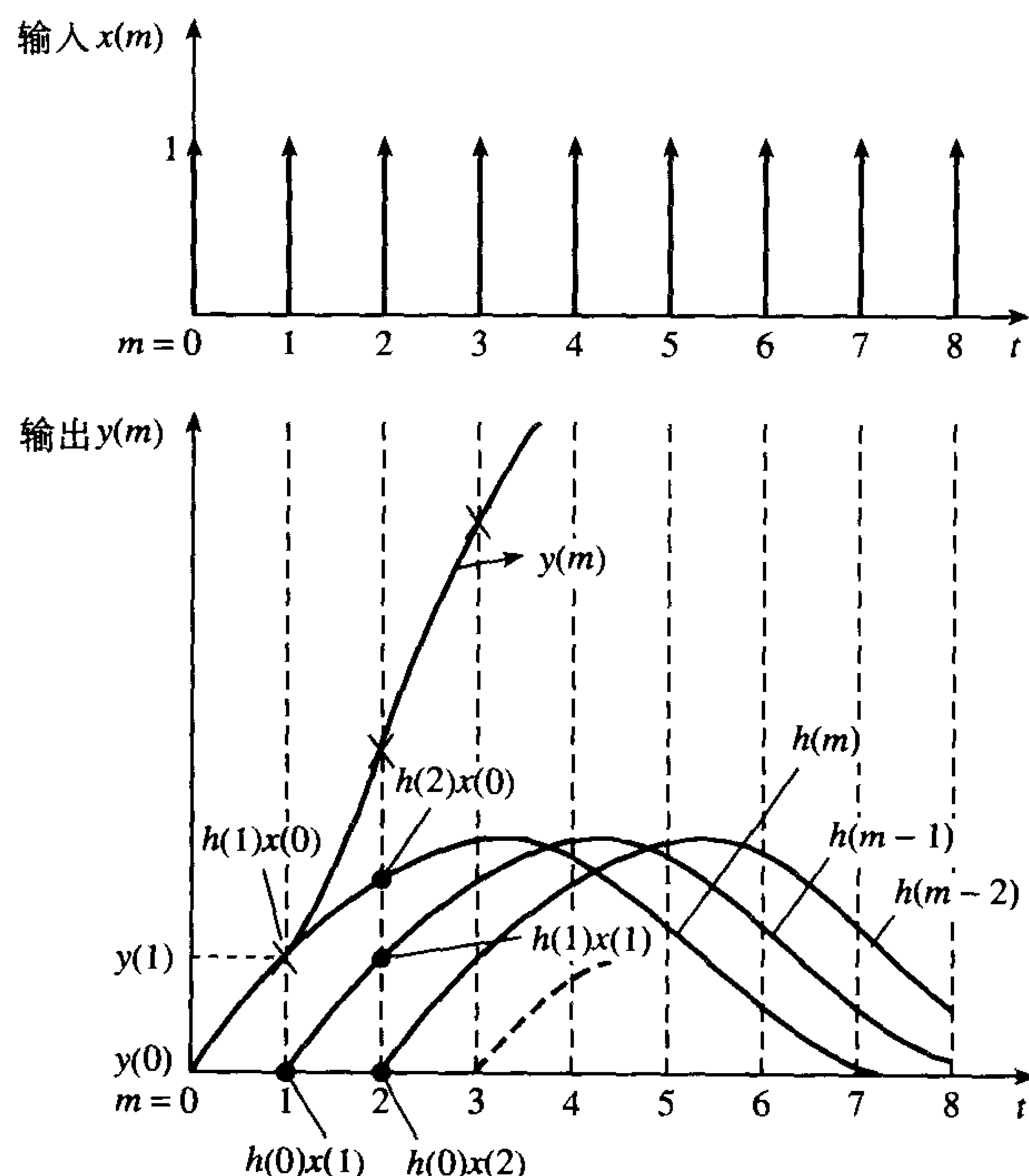


图 5.19　应用的冲激序列和从各个冲激响应导出的系统响应

在抽样瞬间 $m = 1$，输出将由 $h(0)x(1)$ 给出，也就是当前输入 $x(1)$ 的影响，加上在抽样瞬间 $m = 0$ 加入的输入的延迟影响 $h(1)x(0)$，即

$$y(1) = h(1)x(0) + h(0)x(1)$$

类似地，序列的输出将由下式给出：

$$\begin{aligned} y(2) &= h(2)x(0) + h(1)x(1) + h(0)x(2) \\ y(3) &= h(3)x(0) + h(2)x(1) + h(1)x(2) + h(0)x(3) \\ &\vdots \\ y(n) &= h(n)x(0) + h(n-1)x(1) + \ldots + h(0)x(n) \end{aligned} \tag{5.90}$$

如果系统是线性的，输出仅可以用这种形式写成以前输入的影响的线性和。5.90式描述了一个一阶线性系统的输出。

检查上面的表达式，我们会发现，输出是通过输入序列和对应的时间反转的冲激响应函数的点相乘得到的。此外，由于5.90式可以等价地写成

$$y(n) = h(0)x(n) + h(1)x(n-1) + \ldots + h(n)x(0) \tag{5.91}$$

所以输出可以看作为冲激响应函数和时间反转输入序列的对应点对的内积。因此卷积和等价于一个序列和另一个时间反转的序列的互相关。

5.90 式和 5.91 式可以用紧凑形式写为

$$y(n)=\sum_{m=0}^{n}h(n-m)x(m) \tag{5.92}$$

以及

$$y(n)=\sum_{m=0}^{n}h(m)x(n-m) \tag{5.93}$$

上两式的右边称为输入函数和冲激响应函数的卷积和，我们称输出由输入和系统的冲激响应的卷积给定。

通过把上面方程写成如下形式，5.92式和5.93式可以扩展为无限持续时间的波形：

$$y(n)=\sum_{m=-\infty}^{\infty}x(m)h(n-m)=x(n)\circledast h(n) \tag{5.94}$$

以及

$$y(n)=\sum_{m=-\infty}^{\infty}h(m)x(n-m)=h(n)\circledast x(n) \tag{5.95}$$

这是卷积和的一般形式。在这些方程中符号 $\circledast$ 表示卷积运算符。

如果输入是由一个连续的冲激序列组成，那么上面的求和可以用积分代替。所以，例如5.94式变成

$$y(t)=\int_{-\infty}^{\infty}x(\lambda)h(t-\lambda)\,\mathrm{d}\lambda \tag{5.96}$$

我们称为卷积积分。

迄今为止，术语卷积用来描述系统的冲激响应和系统的输入卷积的结果。但是，这个思想可以扩展到任何两个数据序列的卷积，因此以后卷积这个术语可以在更一般的意义上考虑。

下面给出一个例子，两个周期时间序列(4, 3, 2, 1 $\{h(m)\}$)和(1, 2, 3, 4 $\{x(m)\}$)现在进行卷积。图5.20(a)画出了周期性序列(4, 3, 2, 1 $\{h(m)\}$)，图5.20(b)画出了时间反转序列 $h(-m)$，它现在变为(1, 2, 3, 4)。(回顾前面我们知道，卷积和需要一个序列和另一个时间反转序列点与点相乘，也就相当于一个序列和另一个序列的时间反转序列的互相关。)图中也给出了一个宽度等于一个周期的窗口，在这个窗口里对卷积进行计算。显然得到的结果将是周期性的，就像对应的循环卷积的情形一样（参见5.2.1节），所以它仅需要在窗口时间段内计算卷积。图5.20(f)给出用来参考的第二个序列(1, 2, 3, 4 $\{x(m)\}$)。

那么，当 $n=0$ 时，5.92式变为

$$y(0)=\sum_{m=0}^{n}h(-m)x(m)$$

它是通过如图5.20(b)和图5.20(f)所示的加窗数据求互相关得到的：

$$y(0)=4\times1+1\times2+2\times3+3\times4=24$$

当 $n=1$ 时，5.92式变为

$$y(1)=\sum_{m=0}^{n}h(1-m)x(m)$$

它是通过如图5.20(c)和图.20(f)所示的加窗数据求互相关得到的：

$$y(1)=3\times1+4\times2+1\times3+2\times4=22$$

类似地，我们可以求出：

$$y(2) = 2\times 1 + 3\times 2 + 4\times 3 + 1\times 4 = 24$$

以及

$$y(3) = 1\times 1 + 2\times 2 + 3\times 3 + 4\times 4 = 30$$

然后输出序列循环重复。这个输出序列在图 5.20(g)里画出。

当波形能很好地在数学上定义时，就可以解析地执行卷积。通过考虑一个类似的例子，再来说明一下图解的步骤，可能会使读者对卷积过程得到更好的理解。

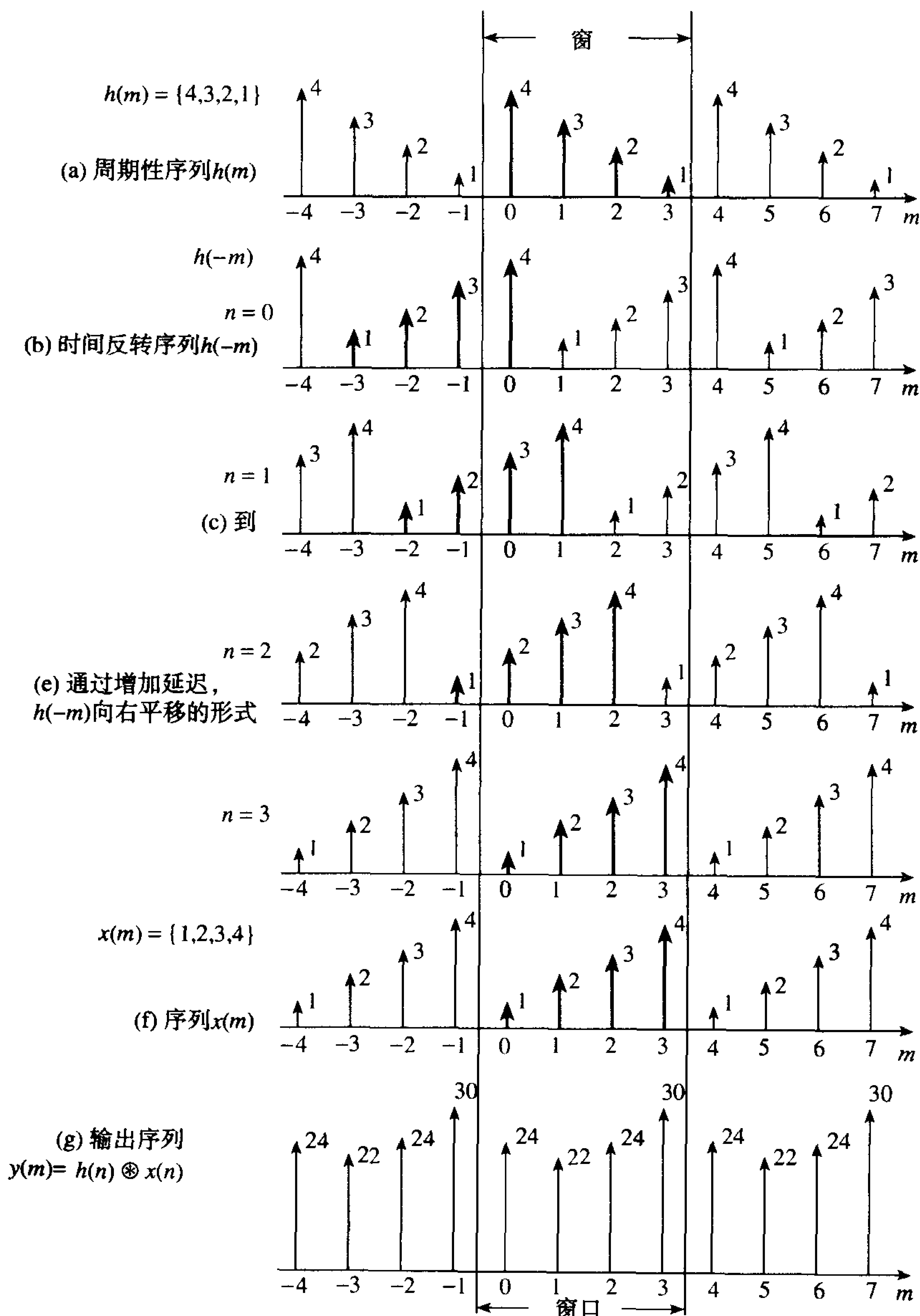


图 5.20　$h(n)$和$x(n)$的卷积$y(m)$

例 5.7　对图 5.21(a)所示的波形 $x(t)$和 $h(t)$进行卷积，并解析地执行。

令卷积积分是

$$y(t) = x(t) \circledast h(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)\,\mathrm{d}\tau \tag{5.97}$$

5.97式对应于5.96式中用τ代替λ后的式子，τ用来表示应用的延迟时间。卷积积分依赖于变量τ，所以图5.21(a)被图5.21(b)代替。

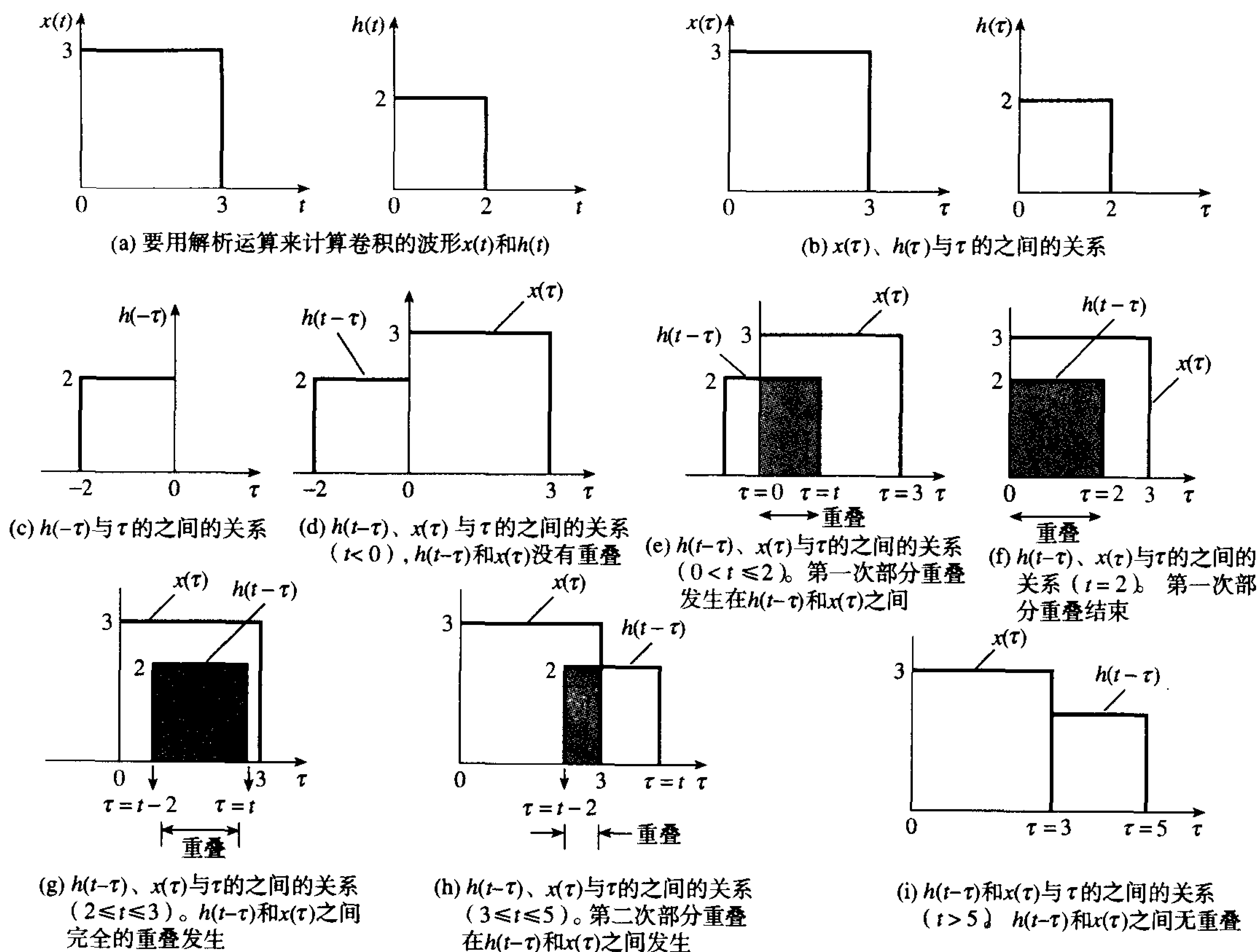


图5.21 $x(t)$和$h(t)$的卷积

现在我们需要对时间进行反转，如图5.21(c)所示。接下来$h(-\tau)$沿着τ的正方向相对于$x(\tau)$平移。那么得到的波形$h(t-\tau)$会在5个不同的几何阶段和$x(\tau)$重叠，如图5.21(d)、图5.21(e)、图5.21(f)、图5.21(g)和图5.21(i)所示。对于每一几何阶段都有一个相应的卷积积分。因此$x(t) \circledast h(t)$是以5个分段连续的区域。

- **第一段** $t<0$，$h(t-\tau)$和$x(\tau)$不重叠（参见图5.21(d)）。因为函数不重叠，所以对于所有的t，$x(\tau)h(t-\tau)=0$，这部分对卷积积分没有贡献。
- **第二段** $0<t\leqslant 2$，$h(t-\tau)$和$x(\tau)$有部分重叠发生（参见图5.21(e)）。在这个范围内

$$y(t) = \int_{\tau=0}^{\tau=t} x(\tau)h(t-\tau)\,\mathrm{d}\tau = \int_{\tau=0}^{\tau=t} (3)\times(2)\,\mathrm{d}\tau$$
$$y(t) = 6[\tau]_0^t = 6t, \quad 0<t\leqslant 2 \tag{5.98}$$

当$t=2$时，这个几何段结束，如图5.21(f)所示。

- **第三段** $2\leqslant t\leqslant 3$，此时$h(t-\tau)$和$x(\tau)$完全重叠（如图5.21(g)所示）。在这个t的范围内，

$$y(t)=\int_{\tau=t-2}^{t}(3)\times(2)\,\mathrm{d}\tau=6[\tau]_{t-2}^{t}$$
$$y(t)=6(t-t+2)=12,\quad 2\leqslant t\leqslant 3 \tag{5.99}$$

● **第四段** $3\leqslant t\leqslant 5$，这是另外一种类型的重叠，如图 5.21(h)所示：

$$y(t)=\int_{\tau=t-2}^{\tau=3}(3)\times(2)\,\mathrm{d}\tau=6[\tau]_{t-2}^{3}=6(5-t)=30-6t \tag{5.100}$$

● **第五段** $t>5$。如图 5.21(i)所示，这是第二次没有重叠，所以它对卷积积分也没有贡献。

因此第二段到第四段对卷积积分有贡献，对于这三段对应的每个区域的卷积积分有不同的表达式，总结如下：

$0<t\leqslant 2 \qquad y(t)=6t$

$2\leqslant t\leqslant 3 \qquad y(t)=12$

$3\leqslant t\leqslant 5 \qquad y(t)=30-6t$

根据这些表达式，可以画出 $y(t)$与 t 的关系图，如图 5.22 所示。

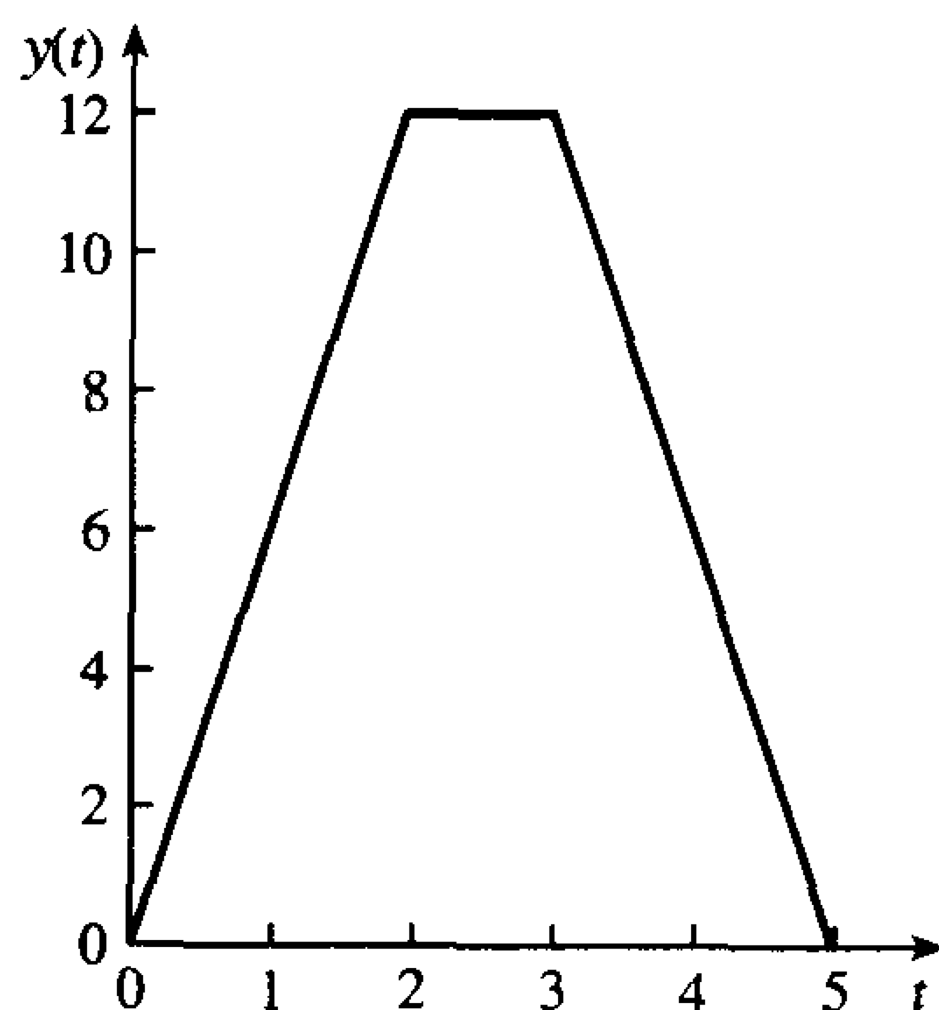


图 5.22 卷积 $y(t)=x(t)\circledast h(t)$对比于 t

我们再重复 5.94 式和 5.96 式一次，这样讨论比较方便：

$$y(n)=\sum_{m=-\infty}^{\infty}x(m)h(n-m)=x(n)\circledast h(n) \tag{5.94}$$

以及

$$y(t)=\int_{-\infty}^{\infty}x(\lambda)h(t-\lambda)\,\mathrm{d}\lambda \tag{5.96}$$

考察这些方程，它提示我们卷积是依时间执行的。这称为时域卷积。我们也知道，频率f处系统的输出分量是 $Y(f)$，它是由下式给出：

$$Y(f)=H(f)X(f) \tag{5.101}$$

其中 $H(f)$是系统在频率f时的频率响应，$X(f)$是输入 $x(t)$的傅里叶变换。也可以证明 $H(f)$是 $h(t)$的傅里叶变换。5.101 式的傅立反叶变换是

$$F^{-1}[Y(f)] = y(t) = F^{-1}[H(f)X(f)] \tag{5.102}$$

把5.96式和5.102式合在一起，可以看出有

$$y(t) = \int_{-\infty}^{\infty} x(\lambda)h(t-\lambda)\,\mathrm{d}\lambda = x(t) \circledast h(t) = F^{-1}[H(f)X(f)] \tag{5.103}$$

因此，可以看出在时间域内两个波形的卷积，等价于两个波形的傅里叶变换的乘积再求傅立反叶变换。这个有用的现象经常用简化形式表述为时域积分等价于频域相乘。

这个关系的对偶性质也存在，也就是频域卷积等价于时域相乘。因而它可以写成（McGillem and Cooper, 1974）：

$$\begin{aligned} Y(\omega) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega-u)H(u)\,\mathrm{d}u = X(f) \circledast H(f) \\ &= F[y(t)] = F[x(t)h(t)] \end{aligned} \tag{5.104}$$

因此，两个时间序列的乘积的傅里叶变换对应于两个序列的傅里叶变换的卷积。这个结果的实际用处是可以解释谱分析之前对数据加窗的影响（参见第11章）。在这个过程中，数字化的数据序列和另外一个由一个窗函数抽样点组成的数据序列点点相乘。这称为加窗，加窗降低了数据能量谱计算的误差。接着计算加窗数据离散傅里叶变换，从而计算出能量谱。目的是获得数据序列的能量谱，但是，从上面式子实际上得到的是数据序列的频谱和加窗序列频谱的卷积。

5.3.1 卷积的性质

(1) 交换律

$$x_1(t) \circledast x_2(t) = x_2(t) \circledast x_1(t) \tag{5.105}$$

我们注意到这等价于

$$\int_{-\infty}^{\infty} x_1(\tau)x_2(t-\tau)\,\mathrm{d}\tau = \int_{-\infty}^{\infty} x_2(\tau)x_1(t-\tau)\,\mathrm{d}\tau$$

(2) 分配律

$$x_1(t) \circledast [x_2(t) + x_3(t)] = x_1(t) \circledast x_2(t) + x_1(t) \circledast x_3(t) \tag{5.106}$$

(3) 结合律

$$x_1(t) \circledast [x_2(t) \circledast x_3(t)] = [x_1(t) \circledast x_2(t)] \circledast x_3(t) \tag{5.107}$$

这些性质既可以通过有关的积分运算来证明，或者根据一个序列和另一个时间反转序列的互相关来考虑卷积。

5.3.2 循环卷积

5.2.1节解释了两个长度不等的周期性序列的相关结果是一个周期等于较短序列的周期性循环序列，这是一个不正确的结果。因为卷积等价于一个序列和第二个反转序列的互相关，所以同样这个结果对卷积来说也是不正确的。因此，和相关一样，在卷积里两个序列长度必须相等。因此，如果序列长度是N_1和N_2，那么对长度为N_1的序列必须增加N_2-1个零，对长度为N_2的序列必须增加N_1-1个零。这样两个序列的长度相等，都等于N_1+N_2-1。在采取了和相关相同的其他预防措施的限制下，可以得到正确的线性卷积。

5.3.3　系统识别

5.95 式给出了一个系统的输入 $x(n)$ 和输出 $y(n)$ 之间的关系式。术语系统识别是指当 $h(n)$ 未知时求出它。如果提供一个测试信号 $x(n)$，测量到输出 $y(n)$，那么除了 5.2.2.4 节讨论的方法之外，$h(n)$ 的确定还有如下的方法。

5.91 式表示 $y(n) = h(0)\,x(n) + h(1)\,x(n-1) + \ldots + h(n)\,x(0)$。当 $n = 0$ 时，$y(0) = h(0)\,x(0)$，所以

$$h(0) = \frac{y(0)}{x(0)} \tag{5.108}$$

现在，展开并整理 5.93，可以得出

$$y(n) = h(n)x(0) + \sum_{m=0}^{n-1} h(m)x(n-m), \quad n \geqslant 1 \tag{5.109}$$

所以

$$h(n) = \frac{y(n) - \sum_{m=0}^{n-1} h(m)x(n-m)}{x(0)} \qquad n \geqslant 1,\ x(0) \neq 0 \tag{5.110}$$

利用 5.108 式和 5.110 式就可以计算出 $h(n)$。

例 5.8　一个测试信号 $x(n) = \{1, 1, 1\}$ 加到一个冲激响应 $h(n)$ 未知的系统。观察到的系统的输出是 $y(n) = \{1, 4, 8, 10, 8, 4, 1\}$，试确定 $h(n)$。

从 5.108 式有

$$h(0) = \frac{y(0)}{x(0)} = \frac{1}{1} = 1$$

利用 5.110 式，

$$h(n) = \frac{y(n) - \sum_{m=0}^{n-1} h(m)x(n-m)}{x(0)}$$

对于 $h(1)$：

$$h(1) = \frac{y(1) - \sum_{m=0}^{0} h(m)x(1-m)}{x(0)} = \frac{y(1) - h(0)x(1)}{x(0)} = \frac{4 - 1 \times 1}{1} = 3$$

对于 $h(2)$：

$$h(2) = \frac{y(2) - \sum_{m=0}^{1} h(m)x(2-m)}{x(0)} = \frac{y(2) - h(0)x(2) - h(1)x(1)}{x(0)}$$

$$= \frac{8 - 1 \times 1 - 3 \times 1}{1} = 4$$

对于 $h(3)$：

$$h(3)=\frac{y(3)-\sum_{m=0}^{2}h(m)x(3-m)}{x(0)}=\frac{y(3)-h(0)x(3)-h(1)x(2)-h(2)x(1)}{x(0)}$$

$$=\frac{10-1\times0-3\times1-4\times1}{1}=3$$

对于 $h(4)$：

$$h(4)=\frac{y(4)-\sum_{m=0}^{3}h(m)x(4-m)}{x(0)}$$

$$=\frac{8-h(0)x(4)-h(1)x(3)-h(2)x(2)-h(3)x(1)}{x(0)}$$

$$=\frac{8-1\times0-3\times0-4\times1-3\times1}{1}=1$$

对于 $h(5)$：

$$h(5)=\frac{y(5)-\sum_{m=0}^{4}h(m)x(5-m)}{x(0)}$$

$$=\frac{y(5)-h(0)x(5)-h(1)x(4)-h(2)x(3)-h(3)x(2)-h(4)x(1)}{x(0)}$$

$$=\frac{4-0-0-0-3\times1-1\times1}{1}=0$$

实际上，$h(n)=0$，$n\geqslant5$。因此 $h(n)=\{1,3,4,3,1\}$。

5.3.4 反卷积

如果一个系统的冲激响应和输出是已知的，那么求未知输入信号的过程称为反卷积。通过一个类似于 5.3.3 节描述的系统识别的过程，可以得到反卷积。展开并用不同的方式整理 5.93 式，得

$$y(n)=h(0)x(n)+\sum_{m=1}^{n}h(m)x(n-m) \tag{5.111}$$

当 $n=0$ 时，$y(0)=h(0)x(0)$。

因此，

$$x(0)=\frac{y(0)}{h(0)} \tag{5.112}$$

整理 5.111 式，得

$$x(n)=\frac{y(n)-\sum_{m=1}^{n}h(m)x(n-m)}{h(0)} \tag{5.113}$$

5.112 式和 5.113 式类似于 5.108 式和 5.110 式，所以 $x(n)$ 的计算类似于 $h(n)$。

例 5.9 假设系统同例 5.8，$h(n)=\{1,3,4,3,1\}$ 和 $y(n)=\{1,4,8,10,8,4,1\}$，计算输入 $x(n)$。

从 5.112 式，

$$x(0) = \frac{y(0)}{h(0)} = \frac{1}{1} = 1$$

从 5.113 式，

$$x(1) = \frac{y(1) - h(1)x(0)}{h(0)} = \frac{4 - 3 \times 1}{1} = 1$$

$$x(2) = \frac{y(2) - h(1)x(1) - h(2)x(0)}{h(0)} = \frac{8 - 3 \times 1 - 4 \times 1}{1} = 1$$

$$x(3) = \frac{y(3) - h(1)x(2) - h(2)x(1) - h(3)x(0)}{h(0)} = \frac{10 - 3 \times 1 - 4 \times 1 - 3 \times 1}{1} = 0$$

对于 $n \geqslant 3$，$x(n) = 0$。

因此 $x(n) = \{1, 1, 1\}$，和例 5.8 用的值是一致的。

5.3.5　盲反卷积

当系统的冲激响应未知时，从输出的信号来确定输入信号的过程称为盲反卷积。获得盲反卷积的方法在下面描述，这个方法基于 Bell 和 Sejnowski（1995）的研究而发展起来。这个问题和它的图解如图5.23所示。在图.23(a)中要求的未知信号源$x(n)$通过一个冲激响应是$h(n)$的系统，产生可测量的输出信号$f(n)$。$f(n)$是 $h(n)$和 $x(n)$的卷积 $h(n) \otimes x(n)$，它是 $x(n)$的延迟形式的变形。现在我们的目的是计算一个信号$u(n)$，它是$x(n)$的一个好的近似。因此，要求一个因果滤波器$w(n)$，当它和$f(n)$卷积时产生 $u(n)$，如图 5.23(b) 所示。一个简单的滤波器将是如图 5.24（参见图 1.4(a)）所示的横向滤波器。这个滤波器的输出是

$$u(n) = \sum_{m=0}^{L-1} w(m)f(n-m)$$

也可以用矩阵表述为

$$\mathbf{U} = \mathbf{WF} \tag{5.114}$$

其中 $\mathbf{U} = \{u(0), u(1), \ldots, u(N)\}^T$，

$$\mathbf{W} = \left\{\begin{array}{cccccc} w(L) & 0 & \cdots & 0 & \cdots & 0 \\ w(L-1) & w(L) & \cdots & 0 & \cdots & 0 \\ \vdots & \cdots & \cdots & \vdots & \cdots & \vdots \\ w(1) & w(2) & \cdots & w(L) & \cdots & 0 \\ \vdots & \cdots & \cdots & \vdots & \cdots & \vdots \\ 0 & \cdots & \cdots & w(1) & \cdots & w(L) \end{array}\right\}$$

$\mathbf{F} = \{f(0), f(1), \ldots, f(N)\}^T$，$N$是时间序列的项数。

Bell 和 Sejnowski（1995）利用信息最大化法则推导了一个自适应计算 $\mathbf{W}$ 中的加权值的算法。因此调整它们以减少$u(n)$点之间的统计相关。这称为白化$u(n)$，因为在一个白色噪声序列里抽样是统计独立的。为了达到这个目的，有必要移去高阶统计相关。这是通过将$u(n)$加到一个非线性转移函数 $g[u(n)]$并在它的输出 $y(n) = g[u(n)]$中使信息最大化来实现的。更新权值的公式是

$$\Delta w(L) \propto \sum_{n=1}^{N}\left(\frac{1}{w(L)} - 2x(n)y(n)\right) \tag{5.115}$$

以及

$$\Delta w(L-j) \propto \sum_{n=j}^{N}(-2x(n-1)y(n)) \tag{5.116}$$

一直用这个算法计算，直到$\Delta w(L)$和$\Delta w(L-j)$的变化很小。接着利用推导出来的延时权值和反卷积出来的数据来实现横向滤波器。

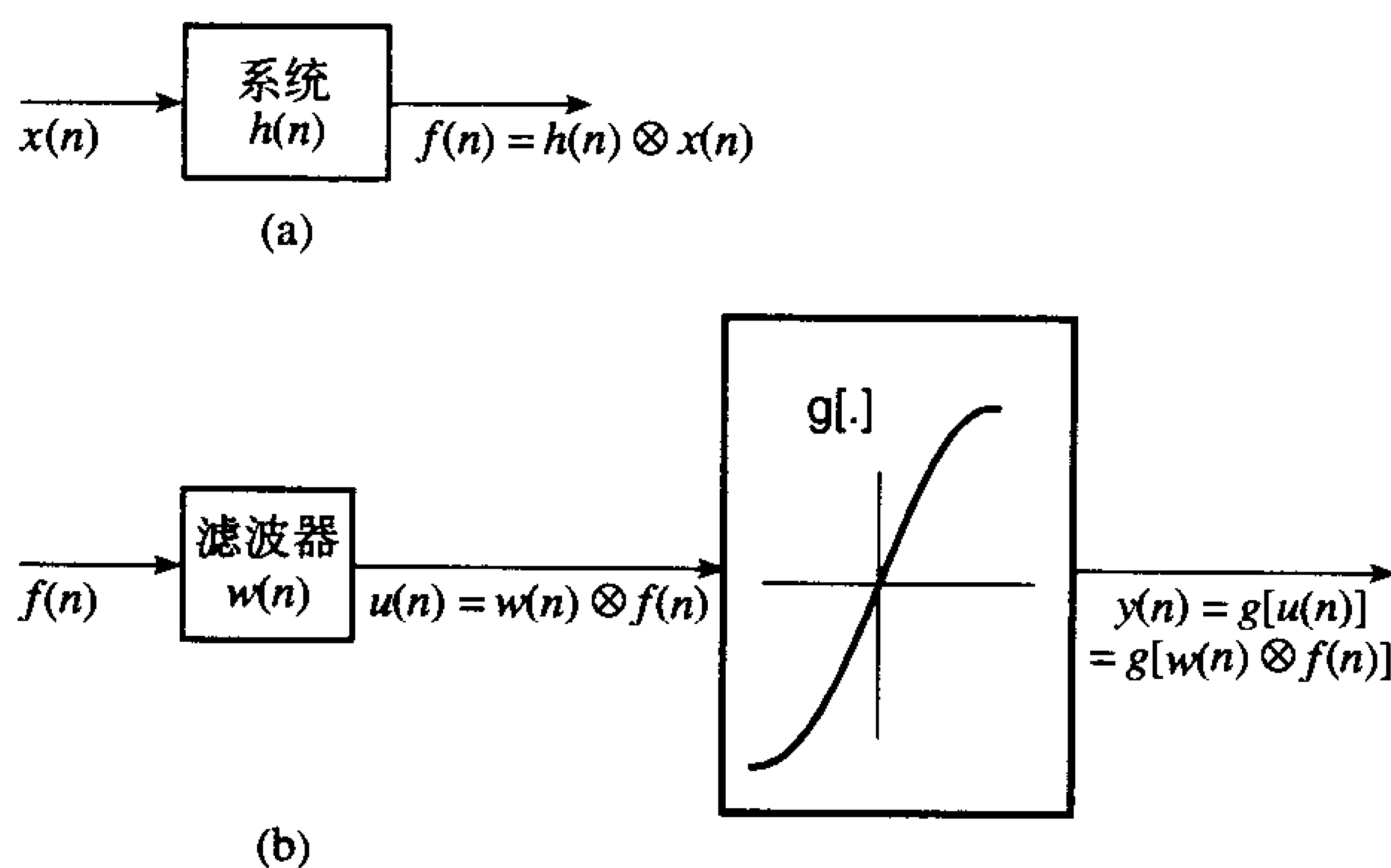


图 5.23　盲反卷积

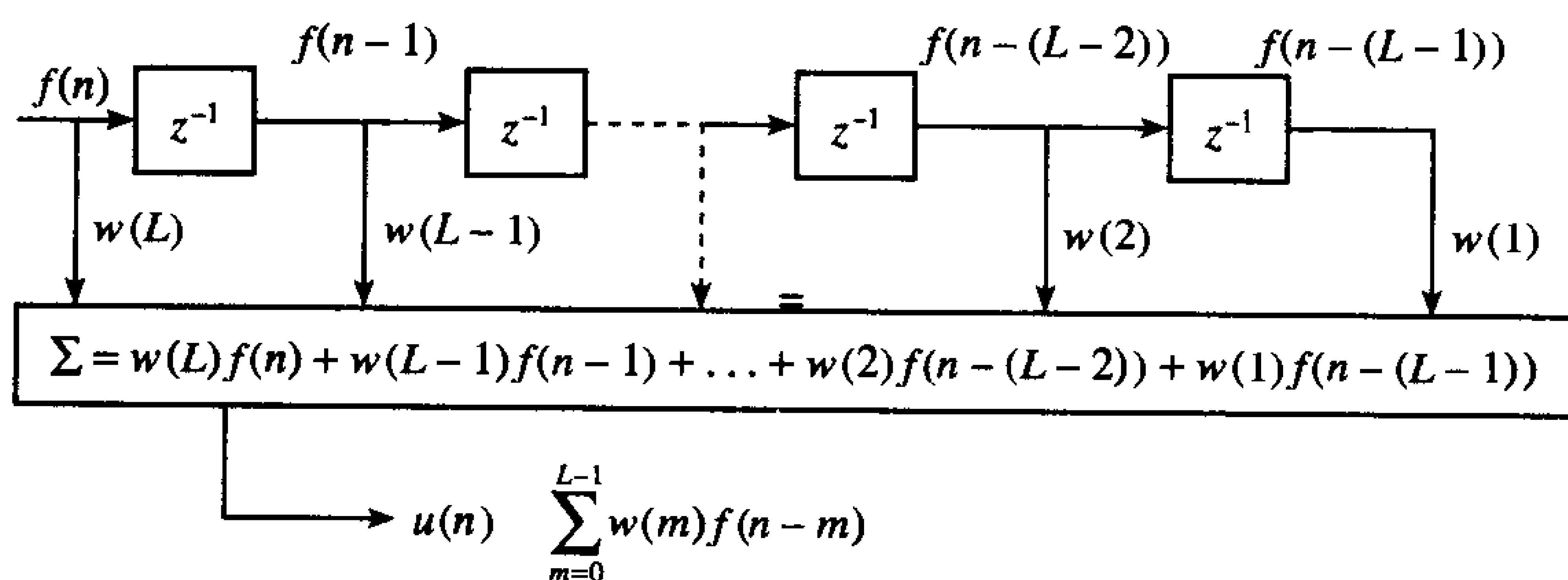


图 5.24　用于盲反卷积的横向滤波器 $w(n)$

5.3.6　快速线性卷积

在 5.2.3 节里，我们证明了相关计算可以用相关定理来加速。在卷积情况下存在一个类似的定理——卷积定理。因此，用离散术语，对于时域，

$$x_1(l) \circledast x_2(r) = F_D^{-1}[X_1(k)X_2(k)] \tag{5.117}$$

5.117 式就是卷积定理，其中F_D^{-1}表示离散傅里叶反变换，$X_1(k)$是$x_1(l)$的离散傅里叶变换，$X_2(k)$是$x_2(r)$的离散傅里叶变换。在 5.2.3 节里，$x_1(l)$和$x_2(r)$是长度为N的周期性序列。

卷积定理的证明

这个定理的证明几乎和 5.2.3 节里的相关定理的证明相同。在卷积中，其中的一个数据序列被反转，而在 6.65 式是用它的共轭，所以

$$X_1(k) = \sum_{l=0}^{N-1} x_1(l)\,\mathrm{e}^{\mathrm{j}(2\pi/N)(-lk)} \tag{5.118}$$

而再次应用 5.66 式：

$$X_2(k) = \sum_{r=0}^{N-1} x_2(r)\,\mathrm{e}^{\mathrm{j}(2\pi/N)(-rk)} \tag{5.119}$$

接着，再一次定义$x_3(n)$为一个长度为N的周期性序列，它对应的 DFT 是$X_3(k)$。$X_3(k)$可以写为

$$X_3(k) = X_1(k)X_2(k) \tag{5.120}$$

接着使用 5.2.3 节的过程，导出要求的时域卷积的结果：

$$x_1(l) \circledast x_2(r) = F_D^{-1}[X_1(k)X_2(k)] \quad (5.121)$$

对于频域的卷积，类似的方程如下所示：

$$\frac{1}{N}[X_1(k) \circledast X_2(k)] = F_D[x_1(l)x_2(r)] \quad (5.122)$$

最后的两个方程代表周期性或者循环性的卷积，如 5.3.2 节里描述的那样，通过增加零可以把它们转化成线性表示。

5.3.7　快速线性卷积的计算优势

当被卷积的序列数目足够大时，快速线性卷积方法比直接方法提供了很大的计算速度优势。在这里我们将直接法和快速法要求的乘法数作为它们计算效率的度量。

直接方法所必需的计算量由 5.90 式给出。从这些方程可以看出，为了求两个 N 点序列 $h(n-m)$ 和 $x(m)$ 的线性卷积，必须把 $h(n-m)$ 的每个值乘以 $x(m)$ 的每一个值。因此，$h(n-m)$ 的 N 个值，每一个都要和 $x(m)$ 的 N 个值相乘，这样总共要做 $N \times N = N^2$ 次乘法。

现在根据 5.121 式，考虑快速方法的线性卷积，两个序列同为 N 点。增加必要的零使得每一个序列的长度变为 $2N-1$ 点。假定 $2N-1 \approx 2N$，例如当 $N \geqslant 8$ 时，为了应用基 -2 FFT，给定 N 是 2 的整数幂，也就是 $N = 2^d$，其中 d 是一个整数。对于 N 点 FFT，复数乘法的数目是 $(N/2)\log_2 N$（参见 3.5.3 节），所以对于 $2N$ 点 FFT，要求 $(2N/2)\log_2 2N$ 个或者 $N\log_2 2N$ 次复数乘法。5.121 式要求计算两个 DFT 和一个 DFT 反变换。反变换的计算是应用修正的 DFT（参见 3.6 节）。因而要求计算三个 $2N$ 点 FFT，包括 $3N\log_2 2N$ 个复数乘法。此外，对于 5.121 式的 $2N$ 个值的每一个，需要计算复数乘法 $X_1(k)X_2(k)$，因此复数乘法的数目增加到 $3N\log_2 2N+2N$。现在，每一个具有形式 $(A+jB)(C+jD)$ 的复数乘法要求四个实数乘法：AC，AD，BC，BD。因此需要 $12N\log_2 2N+8N$ 个实数乘法。

这样我们可以得出结论：直接方法要求 N^2 个实数乘法，而快速卷积方法要求 $12N\log_2 2N+8N$ 个乘法。表 5.1 比较了不同的 N 值时，两种方法要求的实数乘法的数目。这个表证明了当序列包含的数据点数超过 128 时，快速卷积法计算要比直接法快，对于包含数据点数超过 1024 的序列，大约快 10 倍。对于直接相关和快速相关，同样的结论也成立。

表 5.1　对于两个 N 点序列卷积所需要的实数乘法的数目

N	直接法	快速卷积	乘法次数比，快速 / 直接
8	64	448	7
16	256	1088	4.25
32	1024	2560	2.5
64	4096	5888	1.4375
128	16 384	13 312	0.8125
256	65 536	29 696	0.4531
512	262 144	65 536	0.250
1024	1 048 576	143 360	0.1367
2048	4 194 304	311 296	0.0742

5.3.8　分段卷积和相关

迄今为止，我们都是假定要卷积（或者相关）的两个函数具有有限持续时间。然而，也有可能不是这种情况。例如输入数据可以被认为是具有无限持续时间的，或者因为它实际上是连续的，或者更有可能是因为可用的存储器没有大到足够存储它们。在这些情况下，通过把输入数据分成独立的几段分步骤来求卷积，对每一个输入段分别计算，然后把结果合并，这是有必要的。用来分段计算的两种方法称为重叠相加（overlap-add）和重叠保留（overlap-save）方法，下面将对此做出描述。

不过，我们首先在两个函数不是从时间原点处开始的情形下考虑如何使计算更有效，由此来引出要介绍的这两种方法。

图 5.25 给出了两个抽样波形 $x(n)$和 $h(n)$以及它们的卷积 $x(n) \circledast h(n) = y(n)$。$x(n)$和 $y(n)$分别起始于抽样点 a 和 b。所以如果 a 和 b 比数据 $x(n)$和 $h(n)$的数目 N_1 和 N_2 大，那么相当多的计算包括零数据。这样的计算可以通过平移波形到原点以减少计算的数目，如图5.26所示。对每一个波形添加零，使得它们都包含相同数目的点数 $N = N_1+N_2-1$，这样它们周期性的卷积就相当于两个波形的线性卷积。应用卷积定理5.117式和FFT算法来执行卷积。把最后得到的卷积结果沿着 n 轴平移到起始点 $n = a+b$（如图 5.26(d)所示），这样就得到正确的结果。在这个图形中假设 $N = 2^d$，其中 d 是一个整数，这样可以利用基 -2 FFT。

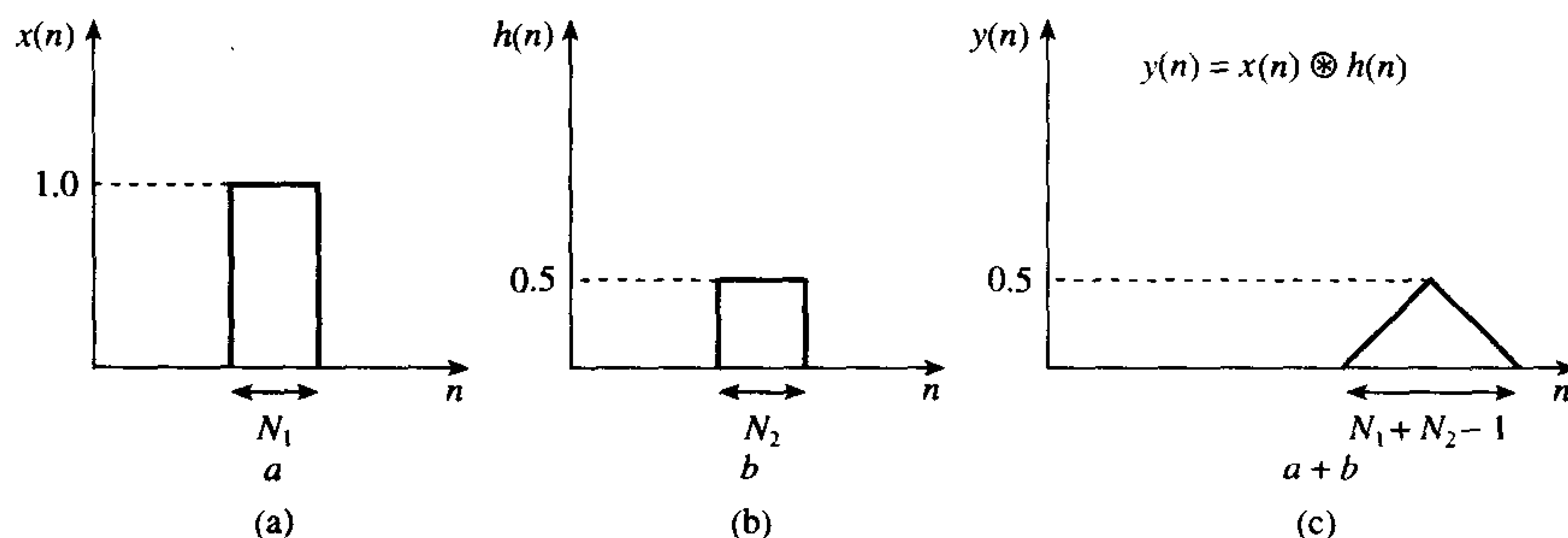


图 5.25　两个不起始于原点的波形 $x(n)$和 $h(n)$的卷积 $y(n) = x(n) \circledast h(n)$

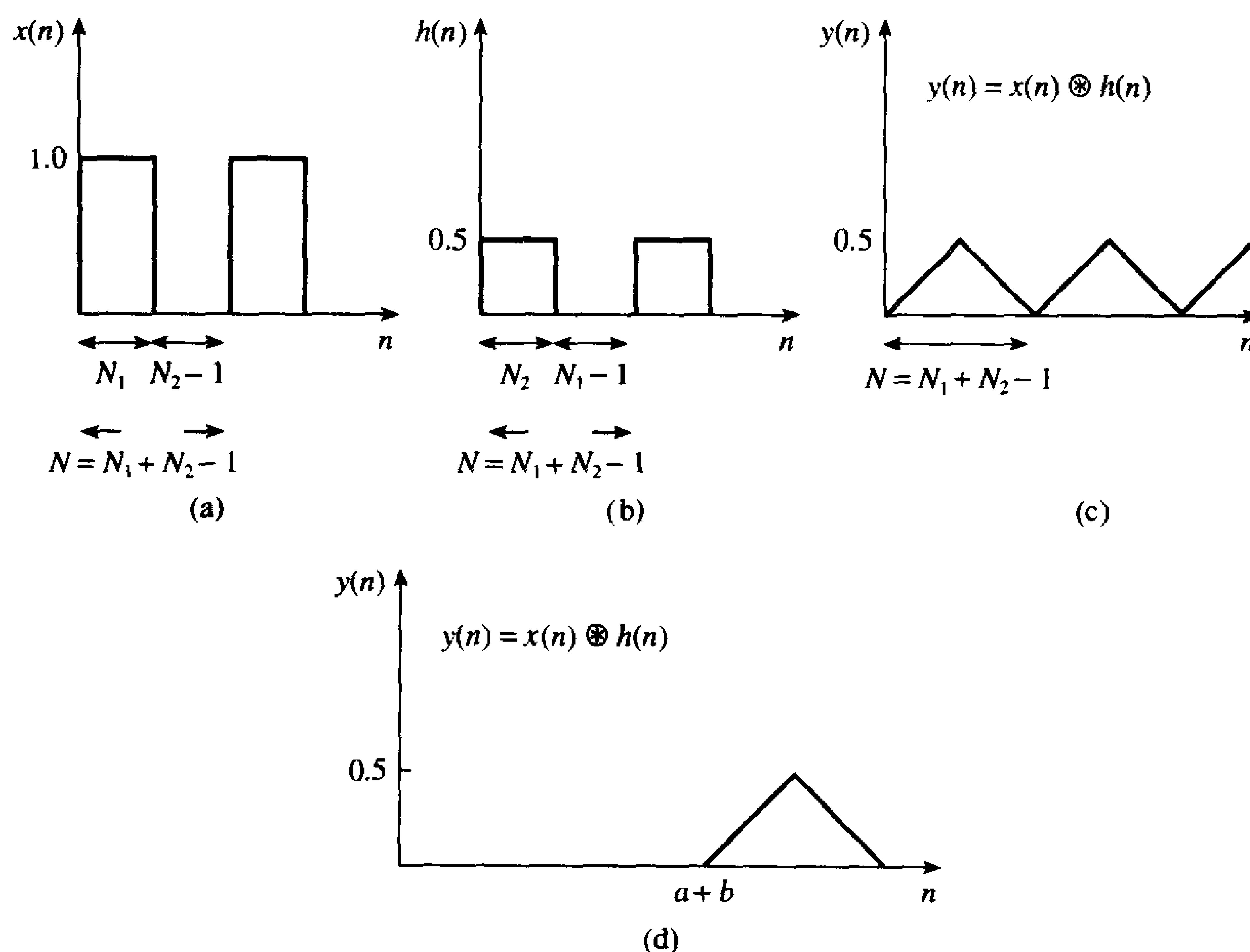


图 5.26　的两个波形开始于原点，通过平移 $x(n)$和 $h(n)$得到两个波形的卷积：(a)给 $x(n)$补 N_2-1 个零；(b)给 $h(n)$补 N_2-1 个零；(c)卷积 $y(n) = x(n) \circledast h(n)$；(d)通过把 $y(n)$沿 n 轴平移到 $n = a + b$ 得到正确的线性卷积

图 5.27 给出了求 $x(n)$和 $h(n)$相关 $r_{xh}(n)$的类似情况。当这些波形变换到原点时，波形要添加零使得 $N = 2^d \geqslant N_1+N_2-1$，并利用相关定理5.77式执行相关运算，得到的波形如图 5.28 所示。这不是图 5.27(c)那样的周期性形式，尽管它具有正确的基本波形。可以通过使 $x(n)$起始于点 $n = N-N_1+1$，而 $h(n)$仍然起始于 $n = 0$ 处，这样最终得到期望的周期性结果，图 5.29 表明这个过程得到了要求的周期性相关函数图5.29(c)。这个结果必需向右平移 $a-b-N+N_1+N_2$ 个数据点，使原点起始于正确的值即 $a-d$ 处（和图 5.29(c)对比）。

现在我们可以扩展我们的讨论：考虑一个无限序列 $x(n)$ 和一个有限序列 $h(n)$ 进行卷积的情况。

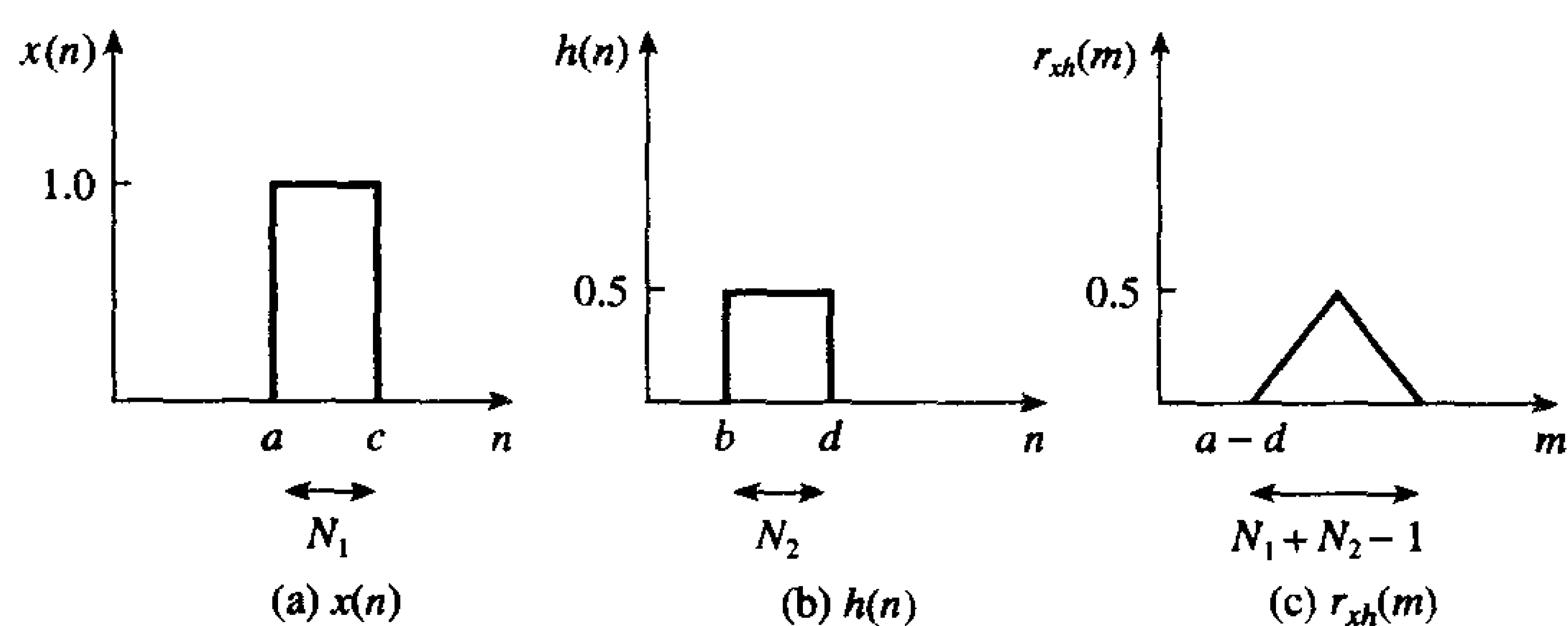


图 5.27　起点不在原点的两个波形 $x(n)$ 和 $h(n)$ 的互相关 $r(m)$

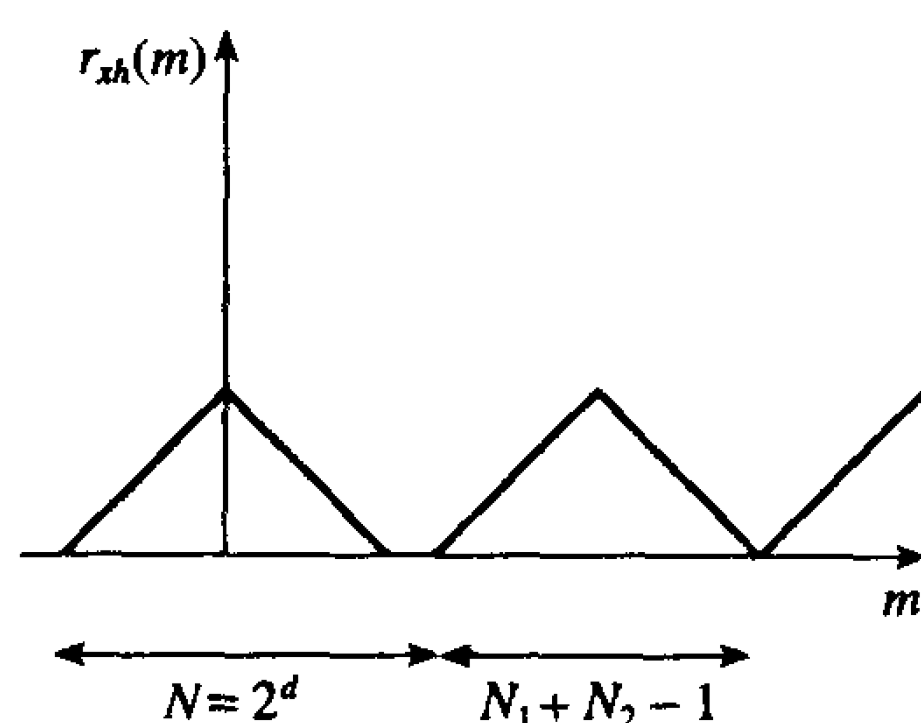


图 5.28　当 $x(n)$ 和 $h(n)$ 被平移到原点时，求 $x(n)$ 和 $h(n)$ 的互相关会得到不正确的值

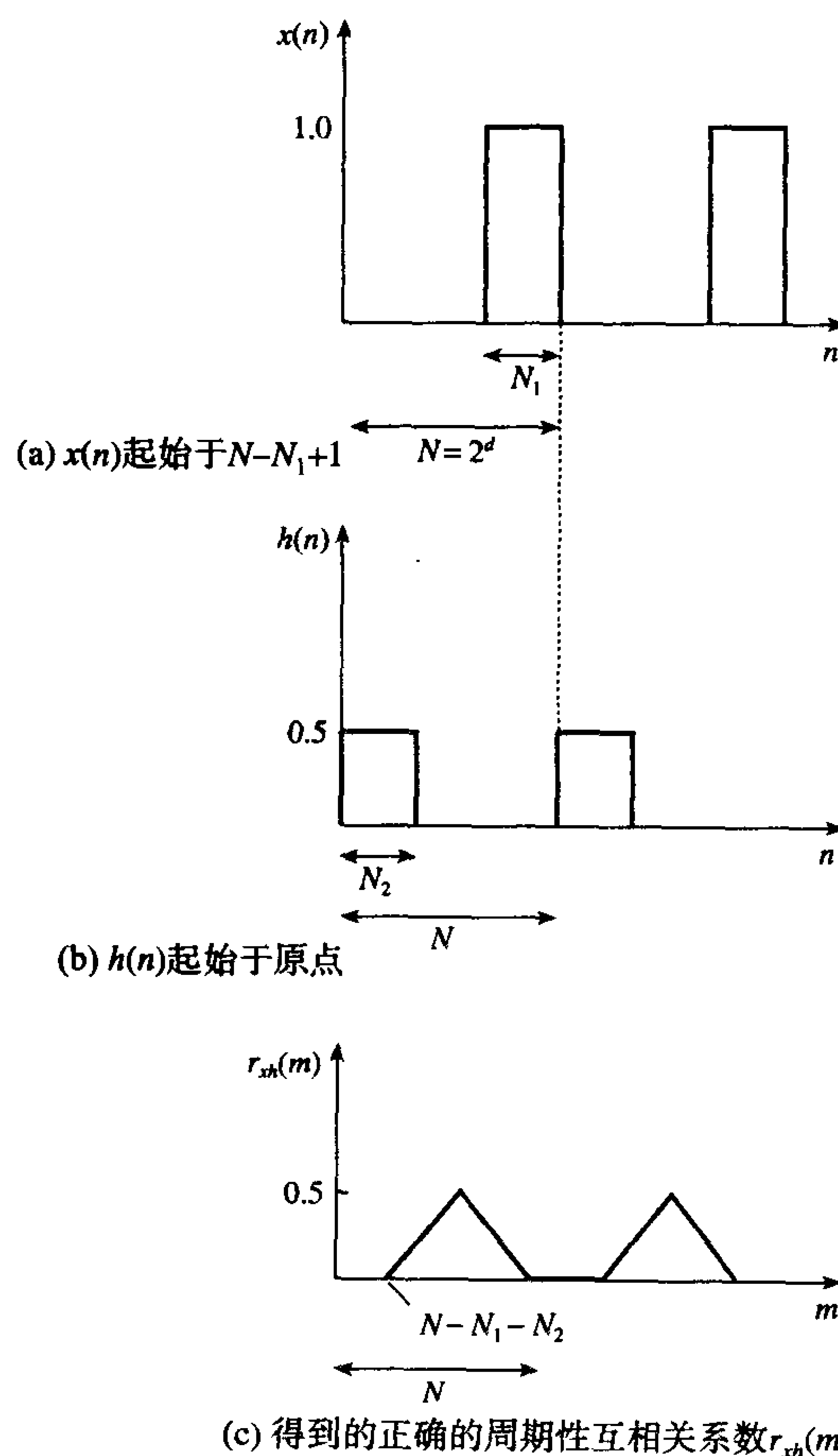


图 5.29　求两个序列 $x(n)$ 和 $h(n)$ 正确的周期性互相关的方法

5.3.9 重叠相加方法

假定$x(n)$被分成相等长度的许多段，每段都是N_1个数据点。现在假设这些分段都是周期性的，它们与$h(n)$进行卷积，$h(n)$的长度是N_2，给$N_2 h(n)$添加N_1-N_2个零，所以每一个序列都是周期性的，且长度为N_1。这样得到的卷积的结果是不正确的，因为要得到一个正确的结果，每一个序列的长度要等于$N = N_1+N_2-1$，但$x(n)$的每一段长度是N_1（不能增加）。考虑把$x(n)$分成长度为N的分段，把后面的N_2-1个数据用零代替，即为前$N-N_2+1 = N_1$个数据补零（参见图5.30）。通过这种方法，具有N_1个数据的序列$x(n)$增加N_2-1个零，具有N_2个数据的序列$h(n)$增加N_1-1个零，每一个序列都包含$N = N_1+N_2-1$个数据，两者正确地进行卷积（参见图5.31）。对于$x(n)$剩下的长度为N的序列，执行同样的过程。因为$x(n)$分段的最后N_2-1个数据被零替代，每个卷积的第一个和最后N_2-1个点是有误差的。但是当把每一个卷积的波形平移到它合适的起始位置（$a+b$）处时，并且卷积的最后N_2-1个点是从与下一个分段卷积重叠的点推导出来时，这些点之和就给出了正确的卷积结果。图5.31解释了这个过程。因此首先要增加足够多的零以消除尾端效应，接着在序列N_1增加了零的地方，卷积的结果重叠，并且精确地加在一起。这就是为什么称为重叠相加方法的原因。

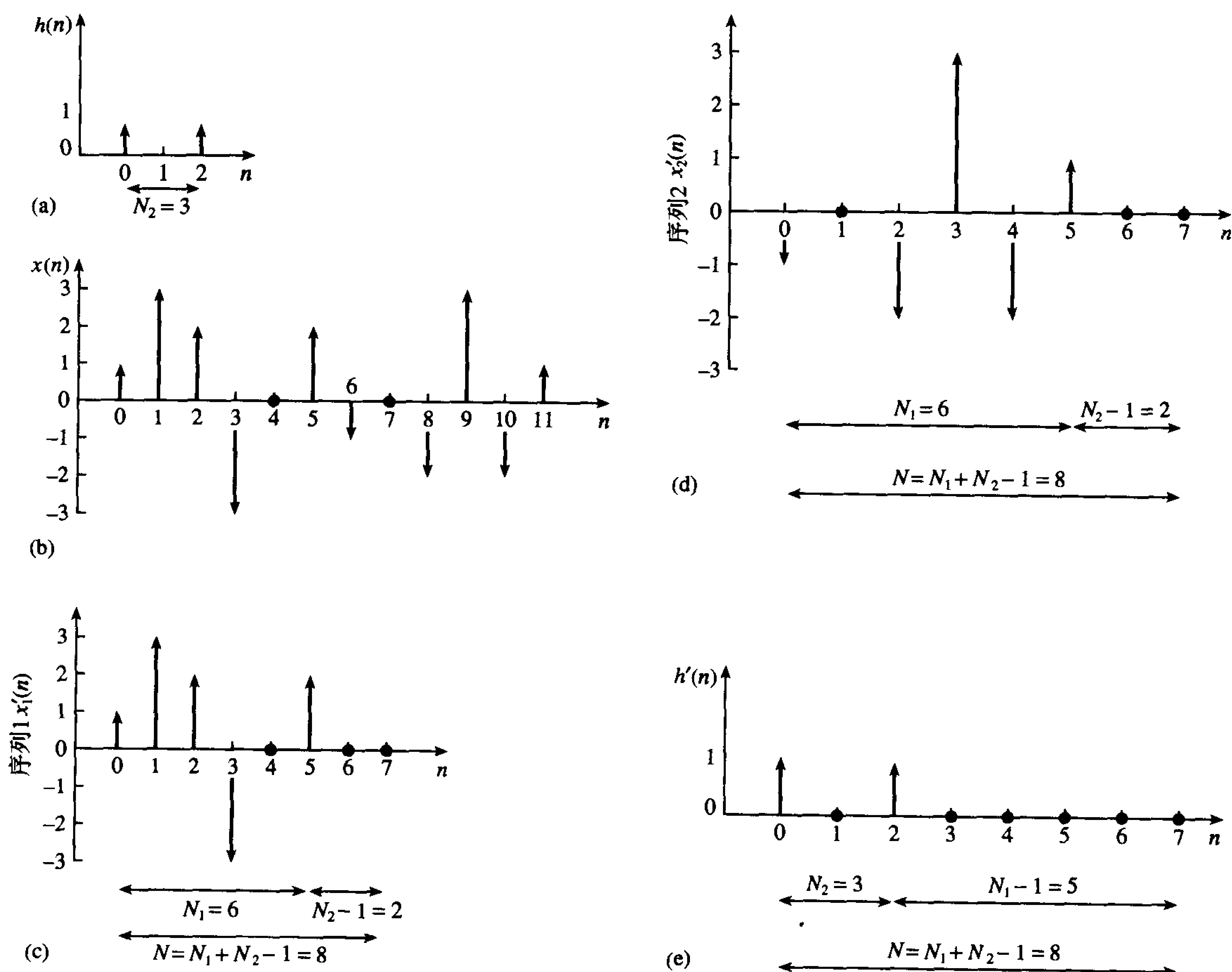


图5.30 卷积的重叠相加方法

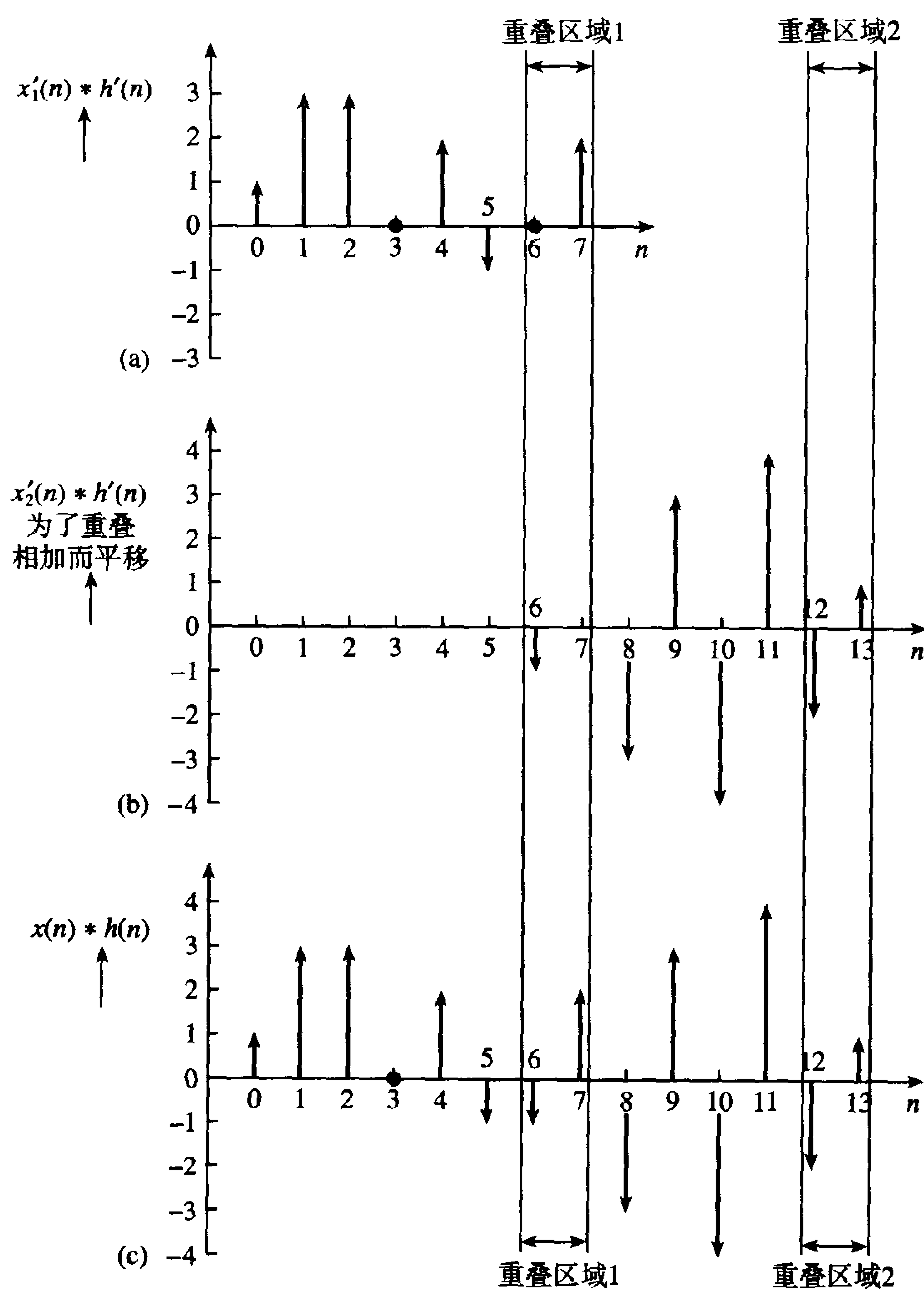


图 5.31　重叠相加卷积和直接卷积的等效性

例 5.10　对于序列 $h(n) = \{1, 0, 1\}$ 和 $x(n) = \{1, 3, 2, -3, 0, 2, -1, 0, -2, 3, -2, 1, \ldots\}$，利用重叠相加方法求卷积。

解:

令 $x(n)$ 分成长度为 $N_1 = 6$ 的许多段，这样 DFT 中点数 N 为 $N_1 + N_2 - 1 = 6+3-1 = 8 = 2^d$，其中 $d = 3$，因此满足线性卷积的要求，也满足基-2 FFT 的应用。

对 $h(n)$ 增加零，得到增加了的序列 $h'(n)$：

$$h'(n) = \{1, 0, 1, 0, 0, 0, 0, 0\}$$

$x(n)$ 的前两个增加的序列是

$$x_1'(n) = \{1, 3, 2, -3, 0, 2, 0, 0\}$$

和

$$x_2'(n) = \{-1, 0, -2, 3, -2, 1, 0, 0\}$$

卷积和$x_1'(n) \circledast h'(n)$的每一项是

$$y_{10} = h_0'x_{10}' = 1$$
$$y_{11} = h_0'x_{11}' + h_1'x_{10}' = 3 + 0 = 3$$
$$y_{12} = h_0'x_{12}' + h_1'x_{11}' + h_2'x_{10}' = 2 + 0 + 1 = 3$$
$$y_{13} = h_0'x_{13}' + h_1'x_{12}' + h_2'x_{11}' = -3 + 0 + 3 = 0$$
$$y_{14} = h_0'x_{14}' + h_1'x_{13}' + h_2'x_{12}' = 0 + 0 + 2 = 2$$
$$y_{15} = h_0'x_{15}' + h_1'x_{14}' + h_2'x_{13}' = 2 + 0 - 3 = -1$$
$$y_{16} = h_0'x_{16}' + h_1'x_{15}' + h_2'x_{14}' = 0 + 0 + 0 = 0$$
$$y_{17} = h_0'x_{17}' + h_1'x_{16}' + h_2'x_{15}' = 0 + 0 + 2 = 2$$

卷积和 $x_2'(n) \circledast h'(n)$ 的每一项是

$$y_{20} = h_0'x_{20}' = -1$$
$$y_{21} = h_0'x_{21}' + h_1'x_{20}' = 0 + 0 = 0$$
$$y_{22} = h_0'x_{22}' + h_1'x_{21}' + h_2'x_{20}' = -2 + 0 - 1 = -3$$
$$y_{23} = h_0'x_{23}' + h_1'x_{22}' + h_2'x_{21}' = 3 + 0 + 0 = 3$$
$$y_{24} = h_0'x_{24}' + h_1'x_{23}' + h_2'x_{22}' = -2 + 0 - 2 = -4$$
$$y_{25} = h_0'x_{25}' + h_1'x_{24}' + h_2'x_{23}' = 1 + 0 + 3 = 4$$
$$y_{26} = h_0'x_{26}' + h_1'x_{25}' + h_2'x_{24}' = 0 + 0 - 2 = -2$$
$$y_{27} = h_0'x_{27}' + h_1'x_{26}' + h_2'x_{25}' = 0 + 0 + 1 = 1$$

上面的两个卷积和分别如图 5.31(a)和图 5.31(b)所示。如果 x_2' 的前 $N_2-1=2$ 个数据和 x_1' 的后 N_2-1 个数据相重叠，并且卷积和相加，那么通过重叠相加方法最后得到的卷积波形的前 12 个数据如图 5.31(c)所示。

上面得到的结果和直接执行卷积得到的结果是等价的。证明过程如下。原始的序列$x(n)$包括12个数据，$h(n)$包括3个数据。为了得到这两个序列的线性卷积，对它们每一个都添加零，使得它们都包含 12+3−1 = 14 个数据。因此序列变为

$$h'(n) = \{1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\}$$

以及

$$x'(n) = \{1, 3, 2, -3, 0, 2, -1, 0, -2, 3, -2, 1, 0, 0\}$$

卷积和中前 9 项为

$$y_0 = h_0'x_0' = 1$$
$$y_1 = h_0'x_1' + h_1'x_0' = 3$$
$$y_2 = h_0'x_2' + h_1'x_1' + h_2'x_0' = 2 + 0 + 1 = 3$$
$$y_3 = h_0'x_3' + h_1'x_2' + h_2'x_1' = -3 + 0 + 3 = 0$$
$$y_4 = h_0'x_4' + h_1'x_3' + h_2'x_2' = 0 + 0 + 2 = 2$$
$$y_5 = h_0'x_5' + h_1'x_4' + h_2'x_3' = 2 + 0 - 3 = -1$$

$$y_6 = h_0' x_6' + h_1' x_5' + h_2' x_4' = -1 + 0 + 0 = -1$$

$$y_7 = h_0' x_7' + h_1' x_6' + h_2' x_5' = 0 + 0 + 2 = 2$$

$$y_8 = h_0' x_8' + h_1' x_7' + h_2' x_6' = -2 + 0 - 1 = -3$$

这个卷积和的各项的确等于图 5.31(c)所示的用重叠相加法得到的各项的值。

通过分段的快速卷积（或者相关）的重叠相加过程如下：

(1) 选择 $x(n)$ 的数据数目 N_1，使得它大于 $h(n)$ 的数目 N_2，即 $N_1 > N_2$，另外选择DFT点数的数目使得 $N = 2^d$，其中 d 是一个整数，并且 $N \geqslant N_1+N_2-1$。通过对数据序列增加必要的零点以满足这些条件。

(2) 把 $x(n)$ 数据增加的分段移向原点。

(3) 对于增加的 $x(n)$ 数据的每一个分段 $x'(n)$，执行快速卷积 $x'(n) \circledast h'(n)$，也就是计算 $X(k)H(k)$，然后求反变换。

(4) 把最后得到的卷积顺序地重叠起来，使它们的最后的值以及前 N_2-1 个值重叠，并把它们相加。

5.3.10　重叠保留方法

让我们再次考虑如图 5.32 所示的卷积 $x(n) \circledast h(n)$，其中 N_2-1 个零添加到 $h(n)$ 上，这样两个序列的长度都是 N_1。通过逐次向右移动 $h(n)$ 一个数据，交叉相乘对应的项，并相加，这样就可以得到这两个序列的线性卷积和。然而，因为两个序列长度都不是 N_1+N_2-1，结果将不是 $x(n) \circledast h(n)$。实际上长度为 N_1 的序列 $x(n)$ 漏掉了 N_2-1 个零。这意味着卷积和的前 N_2-1 项是不正确的，应该舍弃。因此，如果数据 $x(n)$ 被分成连续的长度为 N_1 的段，那么每一段的卷积和的前 N_2-1 个值必须被舍弃。因此卷积 $x(n) \circledast h(n)$ 将包含一个长度为 N_2-1 的缺口的周期性序列。把每一段长度为 N_1 的 $x(n)$ 序列的后 N_2-1 个数据和下一段序列的前 N_2-1 个数据重叠，然后把这些段的前 N_2-1 个数据舍弃，这样就正确地填补了那些缺口。这个过程称为重叠保留方法。

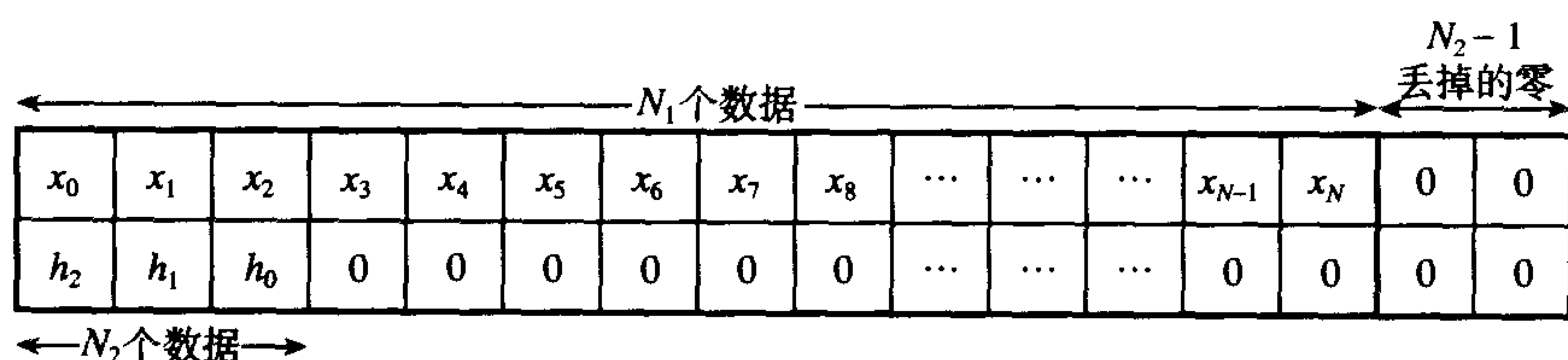


图 5.32　$x(n)$ 和 $h(n)$，其中对 $h(n)$ 增加了 N_2-1 个零

例 5.11　对 5.3.9 节里所给出的那两个序列，应用重叠保留方法求它们的卷积，即

$$h(n) = \{1, 0, 1\}$$

以及

$$x(n) = \{1, 3, 2, -3, 0, 2, -1, 0, -2, 3, -2, 1\}$$

解：

因为 $h(n)$ 有 $N_2 = 3$，所以重叠的数目是 $N_2-1 = 2$。各个分段的重叠如图 5.33 所示。对一个分段的卷积的计算如下。

对于第一段，

$h(n)$	1	0	1									
$x(n)$	1	3	2	−3	0	2	−1	0	−2	3	−2	1
段1	1	3	2	−3								
段2			2	−3	0	2						
段3					0	2	−1	0				
段4							−1	0	−2	3		
段5									−2	3	−2	1

图5.33　卷积的重叠保留法中各段的重叠

$$y_{10} = h_0 x_{10} = 1$$

$$y_{11} = h_0 x_{11} + h_1 x_{10} = 3 + 0 = 3$$

$$y_{12} = h_0 x_{12} + h_1 x_{11} + h_2 x_{10} = 2 + 0 + 1 = 3$$

$$y_{13} = h_1 x_{13} + h_1 x_{12} + h_2 x_{11} + h_3 x_{10} = -3 + 0 + 3 + 0 = 0$$

因此

$$y_1 = \{1, 3, 3, 0\}$$

对于剩下的分段，，记住 $h_1 = h_3 = 0$。我们从第二个分段得到

$$y_{20} = h_0 x_{20} = 2$$

$$y_{21} = h_0 x_{21} = -3$$

$$y_{22} = h_0 x_{22} + h_2 x_{20} = 2 + 0 = 2$$

$$y_{23} = h_0 x_{23} + h_2 x_{21} = 2 - 3 = -1$$

$$y_2 = \{2, -3, 2, -1\}$$

类似地，对第三个分段，

$$y_3 = \{0, 2, -1, 2\}$$

对第四个分段，

$$y_4 = \{-1, 0, -3, 3\}$$

最后，对第五个分段，

$$y_5 = \{-2, 3, -4, 4\}$$

这些结果如表5.2所示，其中说明了每一段序列的前 N_2-1 个结果要舍弃。表中最后一行除了前 N_2-1 个点，都对应了正确的卷积值。

表5.2　例5.11的结果

段1	y_0	~~1 3~~	3 0				
段2	y_1		~~2 3~~	2 −1			
段3	y_2			~~0 2~~	−1 2		
段4	y_3				~~−1 0~~	−3 3	
段5	y_4					~~−2 3~~	−4 4
$x(n) \circledast h(n)$		~~1 3~~	3 0	2 −1	−1 2	−3 3	−4 4

因此，重叠保留方法的过程如下：

(1) 选择 $x(n)$的数据数目，$N_1=2^d$，$x(n)$和 $h(n)$相卷积，给 $h(n)$添加 N_2-1 个零，这样两个序列的长度都是 N_1。
(2) 使两个序列都定位在原点。
(3) 对每一个序列利用 FFT 计算对应的 $X(k)$和 $H(k)$的值。
(4) 计算 $X(k)H(k)$以及它的反变换，它是每一个序列和 $h(n)$的卷积值。
(5) 调整每一个卷积，使它和前一个卷积的最后 N_2-1 个数据重叠。
(6) 把每一个卷积的前 N_2-1 个值舍弃，读出剩下的值，这些值就是对应着正确卷积的值。

5.3.11　分段快速卷积的计算优势

在5.3.8节里我们知道，首先令波形的每一个分段都在原点处卷积，由此可以避免不必要的计算负担，这里我们假定已经是这样处理了。我们进一步假设重叠相加和重叠保留法的计算量是类似的，所以仅需要考虑重叠相加法。假设长度为 N 的序列 $x(n)$被分成 N/N_1 段，每一段具有的长度是 N_1，序列 $h(n)$的长度是 N_2，因而对于线性卷积的序列的长度是 $N^1=2^d\geqslant N_1+N_2-1$。在5.3.7节里已经证明，要执行两个 N_1 点序列的快速卷积要求 $12N^1\log_2 2N^1+8N^1$ 个实数乘法。因此，为了用重叠相加法实现 N 点序列 $x(n)$的快速卷积，要求$(N/N_1)(12N^1\log_2 2N^1+8N^1)=R_m(S)$个实数乘法。这说明了要被卷积的序列的长度 N^1 应该短些，而 $x(n)$数据分段的长度 N_1 应该接近 N^1。理想的情况是 $N^1=2^d=N_1+N_2-1$。原始的 N 点序列要求的实数乘法的数目是 $12N\log_2 2N+8N=R_m(N)$。对于5.3.9节里的例子，表5.3显示了典型的比值是 $R_m(S)/R_m(N)\leqslant 1$，在计算时间上可达到50%的节约。

表5.3　比率 $R_m(S)/R_m(N)$，分段方法的实数乘法的数目和快速卷积直接法的实数乘法的数目的比值

N	N^1	N_1	N/N_1	N_2	$R_m(S)/R_m(N)$	注释
1020	8	6	170	3	0.54	N^1（最好结果）短，$N_1\approx N^1$
1024	256	254	4	3	0.83	$N_1\approx N^1$
1020	128	102	10	3	0.93	
1020	256	204	5	3	1.04	

5.3.12　卷积和相关之间的关系

在卷积中第 n 个输出值由5.93式的卷积和给出：

$$y(n)=\sum_{m=0}^{n}h(m)x(n-m)=h(0)x(n)+h(1)x(n-1)+\ldots+h(n)x(0) \tag{5.123}$$

波形 $h(n)$和 $x(n)$的第 j 个延时的互相关函数的值由5.1式给出，该式经过轻微调整：

$$\begin{aligned}r_{hx}(j)&=\frac{1}{N}\sum_{n=0}^{N-1}h(n)x(j+n)\\&=\frac{1}{N}[h(0)x(j)+h(1)x(j+1)+\ldots+h(N-1)x(j+N-1)]\end{aligned} \tag{5.124}$$

如果 $j=0$，也就是考虑互相关的零延迟情况，那么很容易比较 $y(n)$和 $r_{hx}(j)$。那么5.124式变为

$$\begin{aligned}r_{hx}(0)&=\frac{1}{N}\sum_{n=0}^{N-1}h(n)x(n)\\&=\frac{1}{N}[h(0)x(0)+h(1)x(1)+\ldots+h(N-1)x(N-1)]\end{aligned} \tag{5.125}$$

比较5.123式和5.125式，我们会发现，除了在互相关中$x(n)$序列是反序以外，它们具有类似的形式。因此，把原始序列按时间反转，并且归一化因子$1/N$设为1，那么卷积和互相关等价。这意味着仅需要把一个序列反转，卷积和互相关就可以用相同的计算程序来计算。

5.4 相关和卷积的实现

在考虑相关和卷积这些操作的实现时，我们应该记住，它们两个是密切相关的。两个数据序列既可以求相关，也可以把其中一个数据序列的次序反转来求卷积。另外，对于长数据序列，可以利用快速傅里叶变换法实现快速相关或者卷积来加速运算。当一个数据序列非常长时，利用重叠相加或者重叠保留法比较合适：参见5.3.9节和5.3.10节，以及Brigham(1974)、Strum and Kirk(1988)和DeFatta et al. (1988)。

卷积或相关可以通过FIR滤波器应用FFT来实现。相关或者卷积也可以利用5.2.2节图5.13所示的相关检测器那样的匹配滤波器来实现。对于数字处理，可以用电荷耦合器件（CCD）技术来实现横向滤波器，对于基本延时线结构，这将给出一个数据率可能超过100 MHz的线性相位响应（Grant et al., 1989）。对于模拟处理，可以应用声表面波（SAW）器件通过多抽头延时线来实现（Grant et al., 1989）。这些运算覆盖了从2 MHz到2 GHz的范围。其他的实现包括专用的卷积器和相关器芯片、通用的数字信号处理器、标准微处理器和transputer。后者的一个例子是实时医用系统，它用于从人的EEG的所有16个通道中移去视觉伪像（Jervis et al., 1990）。

快速相关和卷积要求的计算时间如下所示，可以进一步减半（Brigham, 1974）。考虑$x(n)$和$h(n)$的相关。当计算$X(k)$时，把$x(n)$的偶数项存放在FFT的实部，而奇数项存放在FFT的虚部，由此把FFT的长度缩减了一半。那么$(1/N)F_D^{-1}[X(k)H(k)]$的实部给出期望的卷积的偶数项，而虚部给出奇数项。

同样，两个数据序列$x_1(n)$和$x_2(n)$与$h(n)$的卷积可以同时计算。把一个FFT的实部存放$x_1(n)$而虚部放$x_2(n)$，变换得到$X^1(k)$。那么$(1/N)F_D^{-1}[X^1(k)H(k)]$的实部就是$x_1(n) \circledast h(n)$，虚部是$x_2(n) \circledast h(n)$。

5.5 应用实例

5.5.1 相关

例5.12 这个简化了的例子涉及相关理论的应用，它是用来控制空间飞行器的姿态以保证太阳能电池板能一直对着太阳。姿态误差用多个电平脉冲来表示，脉冲的电平间距是$a = 0.2$ mV，脉冲宽度是$T_s = 1$ μs。当存在着一个正的误差时，通过发射一个高度为a的负的脉冲序列来控制姿态误差，如此最初尝试进行控制。只有当误差和控制信号间的相关系数优于−0.5时，才认为满足了控制系统的要求。图5.34(a)给出了三个误差脉冲，而图5.34(b)给出了相应的控制信号脉冲。对于这个例子，我们假定这些脉冲是足够的，不需要考虑大于T_s的延时。这个问题是用来确定系统是否可以看成是满意的。

当$0 \leqslant \tau \leqslant T_s$时，确定是否$|r_{12}(\tau)| > 0.5$，就可以证明系统是否令人满意。通过向右平移控制信号而保持误差信号固定可以求互相关值。这意味着要求$r_{12}(-\tau)$。

而现在

$$r_{12}(-\tau) = \int_{-\infty}^{\infty} v_1(t)v_2(t-\tau)\,\mathrm{d}t$$

其中$v_1(t)$是误差信号，$v_2(t)$是控制信号。

$$r_{12}(-\tau)=\frac{1}{3T_s}\int_{\tau}^{T_s}3a(-a)\,\mathrm{d}t+\frac{1}{3T_s}\int_{T_s}^{2T_s}5a(-a)\,\mathrm{d}t+\frac{1}{3T_s}\int_{2T_s}^{3T_s}4a(-a)\,\mathrm{d}t$$

$$=\frac{a^2}{3T_s}\{[-3t]_{\tau}^{T_s}+[-5t]_{T_s}^{2T_s}+[-4t]_{2T_s}^{3T_s}\}$$

$$=\frac{a^2}{3T_s}(-3T_s+3\tau-10T_s+5T_s-12T_s+8T_s)$$

$$=\frac{a^2}{3T_s}(-12T_s+3\tau)$$

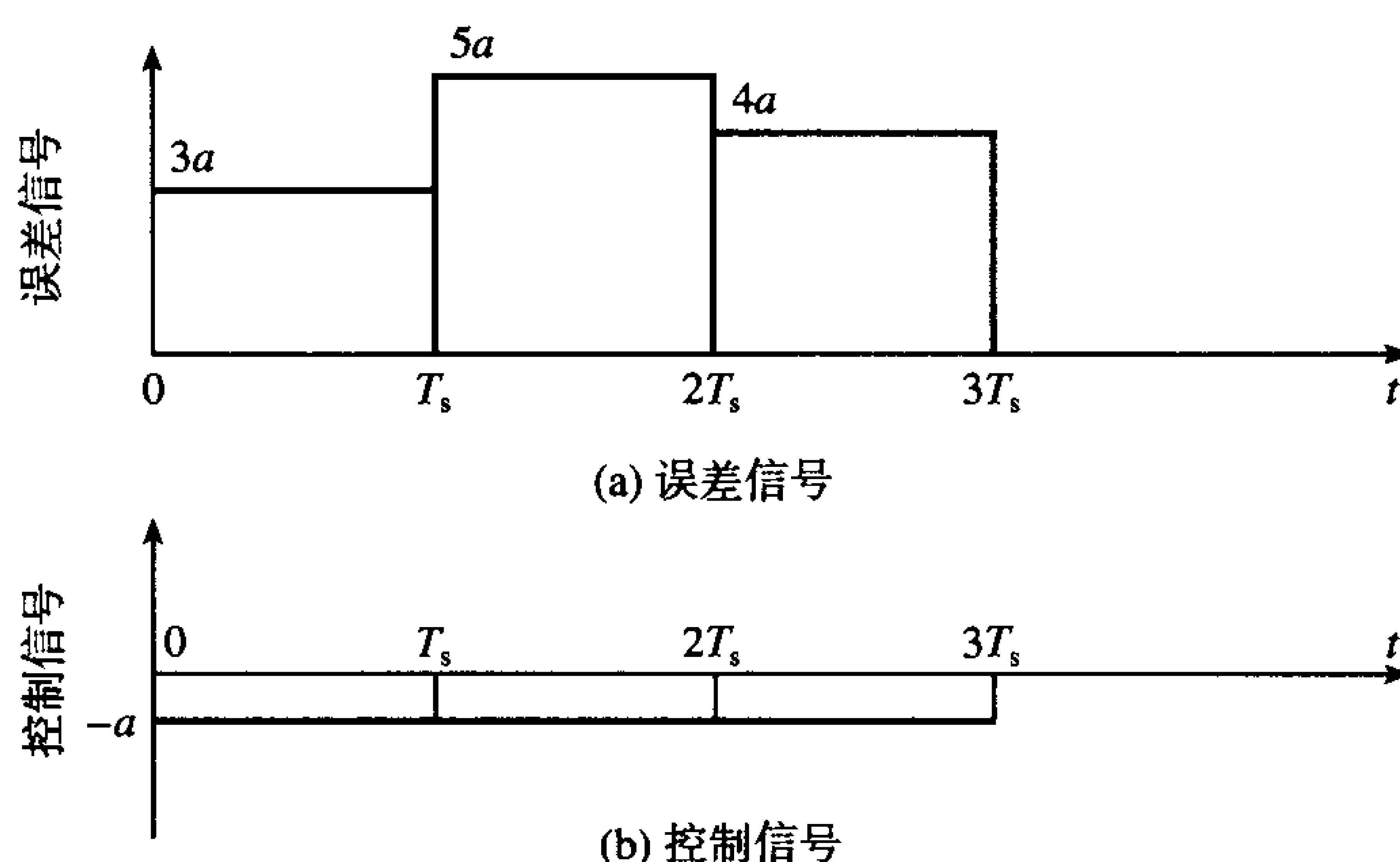


图 5.34　空间飞行器的姿态控制

通过除以下面的归一化因数，把 $r_{12}(\tau)$ 归一化，这样限制值的范围在 $-1\leqslant r_{12}(\tau)\leqslant 1$：

$$\frac{1}{3T_s}\left[\int_{-\infty}^{\infty}v_1^2(t)\,\mathrm{d}t\int_{-\infty}^{\infty}v_2^2(t)\,\mathrm{d}t\right]^{1/2}$$

而

$$\int_{-\infty}^{\infty}v_1^2(t)\,\mathrm{d}t=\int_{0}^{T_s}(3a)^2\,\mathrm{d}t+\int_{T_s}^{2T_s}(5a)^2\,\mathrm{d}t+\int_{2T_s}^{3T_s}(4a)^2\,\mathrm{d}t$$

$$=a^2\{[9t]_0^{T_s}+[25t]_{T_s}^{2T_s}+[16t]_{2T_s}^{3T_s}\}$$

$$=a^2(9T_s+25T_s+16T_s)=50a^2T_s$$

另外

$$\int_{-\infty}^{\infty}v_2^2(t)\,\mathrm{d}t=\int_{0}^{3T_s}(-a)^2\,\mathrm{d}t=a^2[t]_0^{3T_s}=3a^2T_s$$

因此归一化因数是

$$\frac{1}{3T_s}[(50a^2T_s)(3a^2T_s)]^{1/2}=\frac{1}{3T_s}150^{1/2}a^2T_s$$

$r_{12}(-\tau)$ 的归一化表达式是

$$r_{12}^{\mathrm{N}}(-\tau)=\frac{3\tau-12T_s}{150^{1/2}T_s}=\frac{3\tau}{12.25T_s}-\frac{12}{12.25}$$

$$r_{12}^{\mathrm{N}}(-\tau)=0.245\times10^6\tau-0.98$$

当 $\tau=0$ 时，

$$r_{12}^{\mathrm{N}}(0)=-0.98$$

当 $\tau=1\ \mu\mathrm{s}$（允许的最大值）时，

$$r_{12}^{\mathrm{N}}(10^{-6})=-0.735$$

因此，该结果超出了 $|r_{12}^{\mathrm{N}}(-\tau)|>|0.5|$ 这个范围，我们认为它满足准则，能很好地控制空间飞行器的姿态。

例 5.13 为了确定一个声源的距离，需要求一个声呐系统。这个声源是宽带、零均值高斯分布的。系统是由两个间距为 d 的声呐传感器和一个相关联的信号处理系统组成。传感器 T_1 和 T_2 分别接收宽带噪声信号 $q_1(t)$ 和 $q_2(t)=Aq_1(t+\Delta t)$，Δt 是由于从两个传感器到声源路径长度不同而带来的延迟，A 是有关的衰减因数（在这种情况下假设为 $A=1$）。信号处理系统计算两个传感器相等长度的输出的相关函数。

画出在尽可能短的时间里实现相关的简单系统的设计框图，并解释你所依据的原理。

画出传感器输出信号和它们的互相关函数，并简要说明其显著特性。

如果互相关函数的峰值是 10，接收机的带宽是 1 ~ 10 Hz，计算其接收的能量。

解：

系统框图如图 5.35 所示。在这个设计里，通过应用相关定理以及计算所涉及的 FFT，这个系统可以加速相关计算。当数据序列的点数超过 128 个时，这个结构比直接计算相关要快。因此这个系统计算的 $r_{12}(\tau)$ 可以按数字形式表示为

$$r_{12}(j)=F_{\mathrm{D}}^{-1}[F_1(k)F_2^*(k)]$$

系统的输出 $r_{12}(j)$ 是

$$r_{12}(j)=\frac{1}{N}\sum_{n=0}^{N-1}q_1(n)Aq_1(n+\Delta n+j)$$

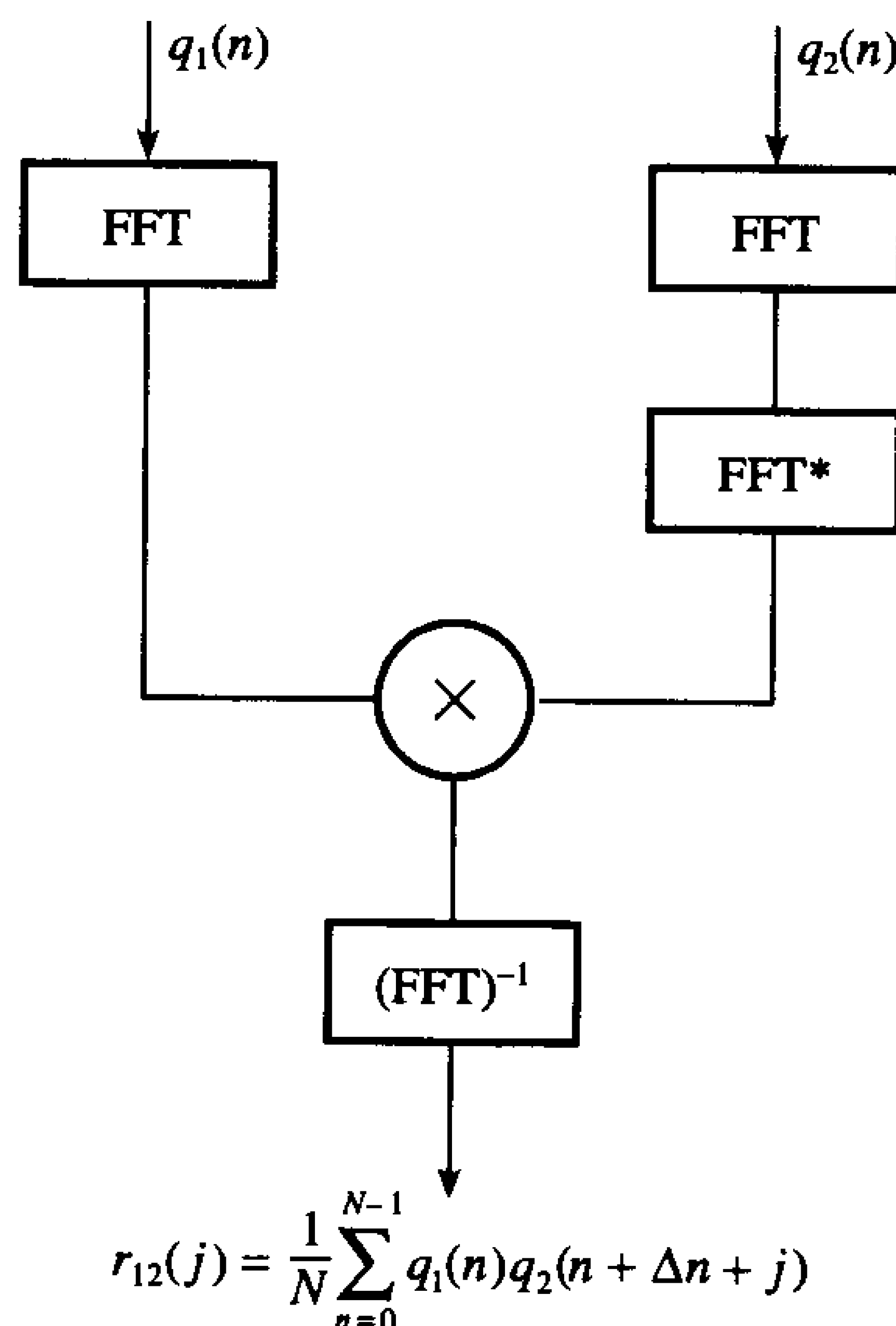


图 5.35 声呐系统流程图

因为 $q_1(n)$和 $q_2(n)$是随机的，当波形在相位上有相对移动时，系统仅产生一个显著的输出。当 $j=-\Delta n$ 时，会发生这种现象。那么输出是

$$\frac{1}{N}\sum_{n=0}^{N-1} q_1^2(n) = P_{\mathrm{AV}},\ \text{平均功率}$$

图 5.36 和图 5.37 给出了波形和它们的互相关函数。

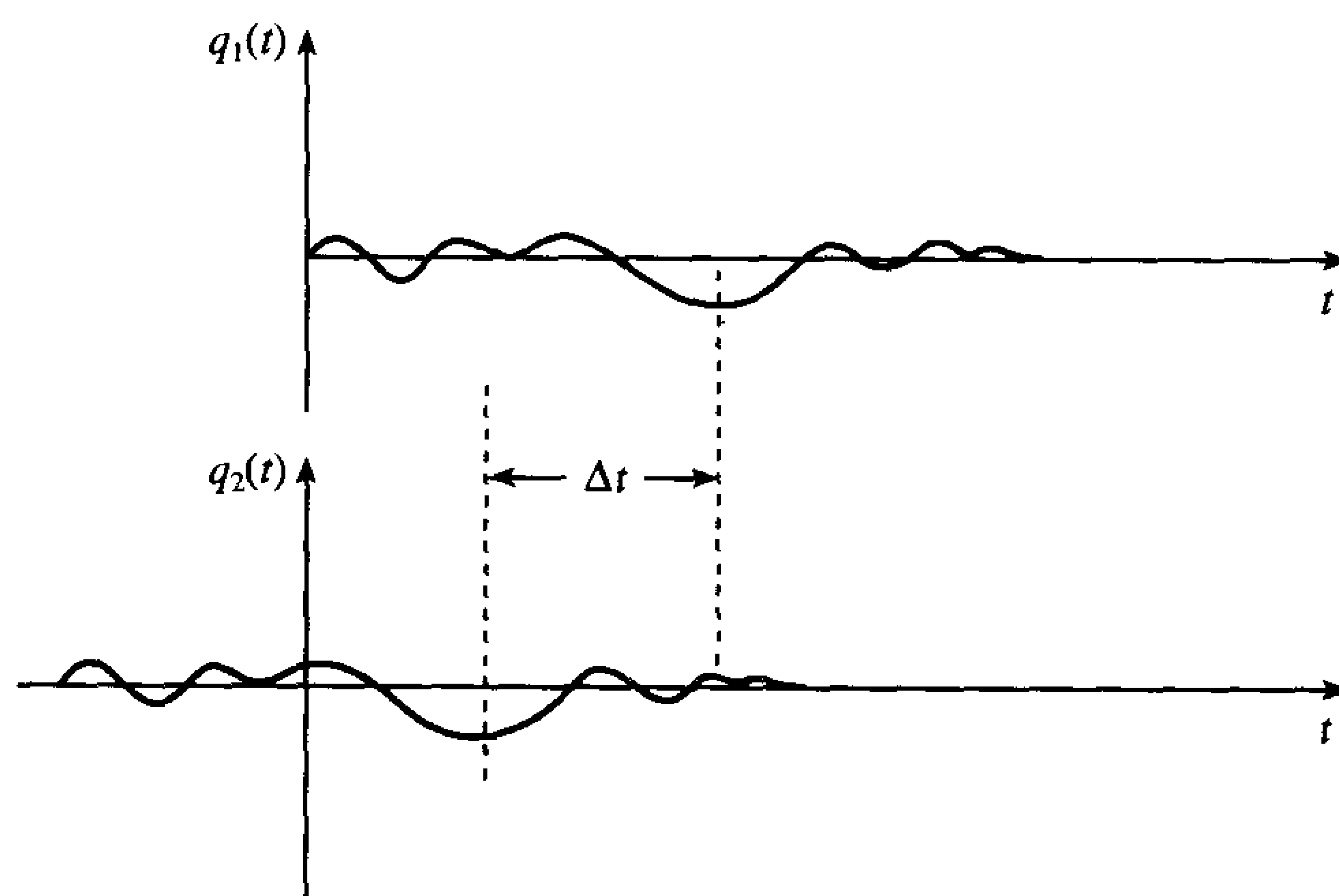


图 5.36 声呐系统检测到的宽带噪声

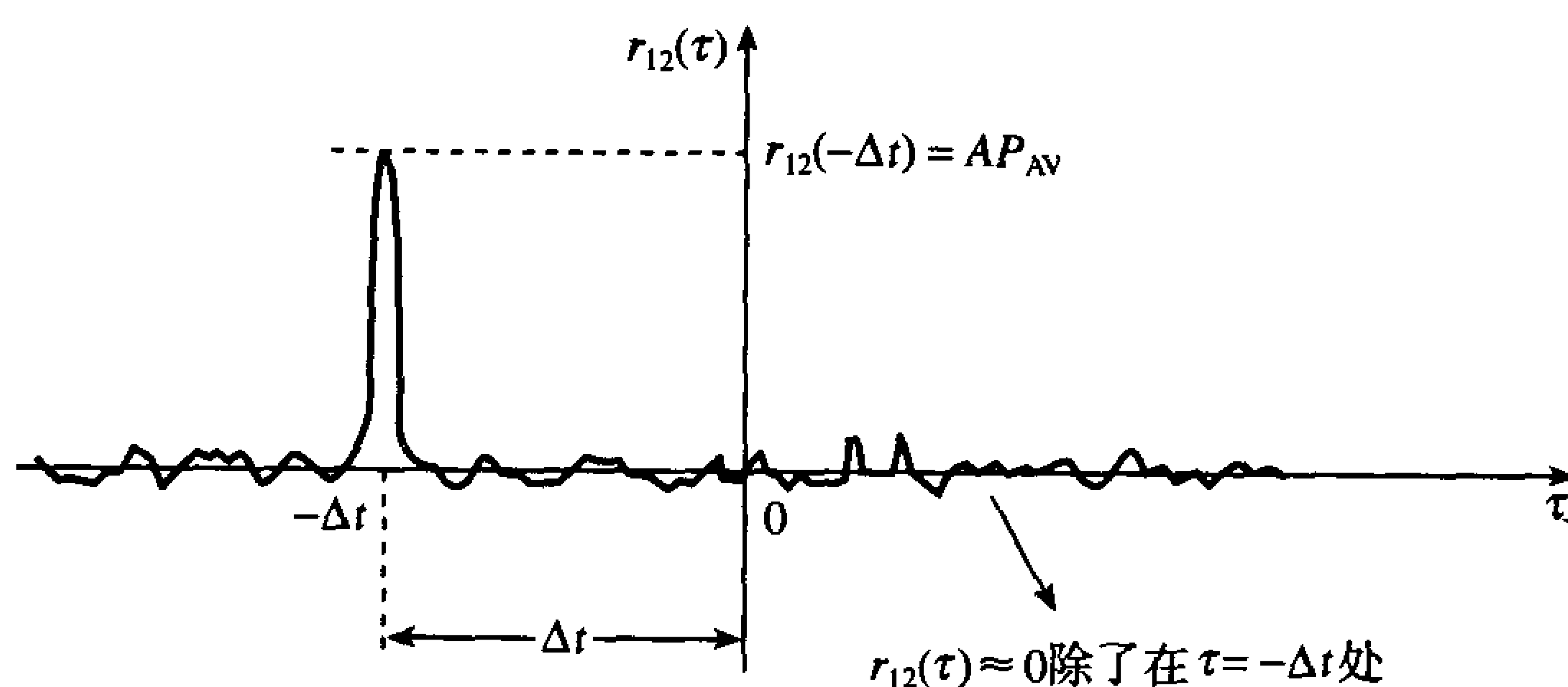


图 5:37 声呐系统信号检测的互相关函数

两个波形的互相关是

$$r_{12}(\tau) = \frac{1}{T}\int_{-T/2}^{T/2} q_1(t)q_2(t+\tau)\,\mathrm{d}t$$

把 $q_2(t)$代入，上式变为

$$r_{12}(\tau) = \frac{1}{T}\int_{-T/2}^{T/2} q_1(t)Aq_1(t+\Delta t+\tau)\,\mathrm{d}t$$

这可以写为

$$r_{12}(\tau) = \frac{A}{T}\int_{-T/2}^{T/2} q_1(t)q_1(t+\tau')\,\mathrm{d}t,\quad \text{其中}\quad \tau' = \Delta t + \tau$$

可以看出被积函数在幅度上等价于 $q_1(t)$零延时的自相关，因此代表这个信号的功率 P_{AV}。因此

$$r_{12}(\tau) = AP_{AV}\delta(t + \Delta t)$$

其中δ表示冲激函数。可以看出$r_{12}(\tau)$的幅度是AP_{AV}。因此$AP_{AV} = 10$。

我们可以首先将维纳－辛钦（Wiener-Khintchine）定理应用到接收的能量谱密度，可以得到在接收机要求的带宽内接收的能量。这个定理是

$$\begin{aligned} G_E(f) &= F_D[r_{12}(\tau)] \\ &= F_D[AP_{AV}\delta(t + \Delta t)] = AP_{AV}\,e^{j\omega\Delta t} \end{aligned}$$

所以，$|G_E(f)| = AP_{AV} = 10\ \mathrm{J\ Hz^{-1}}$。这个信号带宽是$(10-1)\ \mathrm{Hz} = 9\ \mathrm{Hz}$。因此接收到的能量是$10 \times 9 = 90\ \mathrm{J}$。

5.5.2 卷积

5.5.2.1 FIR和IIR滤波器

横向滤波器运算，不管是FIR还是IIR，都提供了相当好的卷积应用实例（Stremler, 1982; DeFatta et al., 1988）。它们可以用来设计对序列求卷积或者执行更一般的数字滤波，例如用于图像处理的二维滤波（Grant et al., 1989）、噪声抑制、图像增强和模式识别等。

考虑一个线性时不变（LTI）系统，对它的描述如下：

$$y(n) = \sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{L} b_k x(n-k) \qquad (5.126)$$

其中$y(n)$代表输出序列，$x(n)$代表输入序列。可以看出输出依赖于当前的输入以及过去的输入和输出。a_k和b_k是实常数，N是方程的阶数，代表必须考虑的以前的输出数目。

因为当前的输出依赖于以前的输出，因此系统是递归的。如果系统的输出仅依赖于以前的输入，那么称它为非递归的，可以通过下式描述它：

$$y(n) = \sum_{k=0}^{L} b_k x(n-k) \qquad (5.127)$$

这个方程是一个横向滤波器（或者说多抽头延迟线）的描述。

图5.38给出了5.127式所示的系统的图表描述。和中的项表示系统的输出，它是对输入值延迟并加权求和。现在我们将要证明，这些权值对应于系统的冲激响应。假设输入$x(n)$是一个单位冲激$\delta(n)$，其中

$$x(n) = \delta(n) = \begin{cases} 1, & n = 0, \quad 即\ x(0) = 1 \\ 0, & n \neq 0, \quad 即\ x(n \neq 0) = 0 \end{cases}$$

对应的输出是冲激响应$h(n)$。把连续的输入值代入5.127式得到

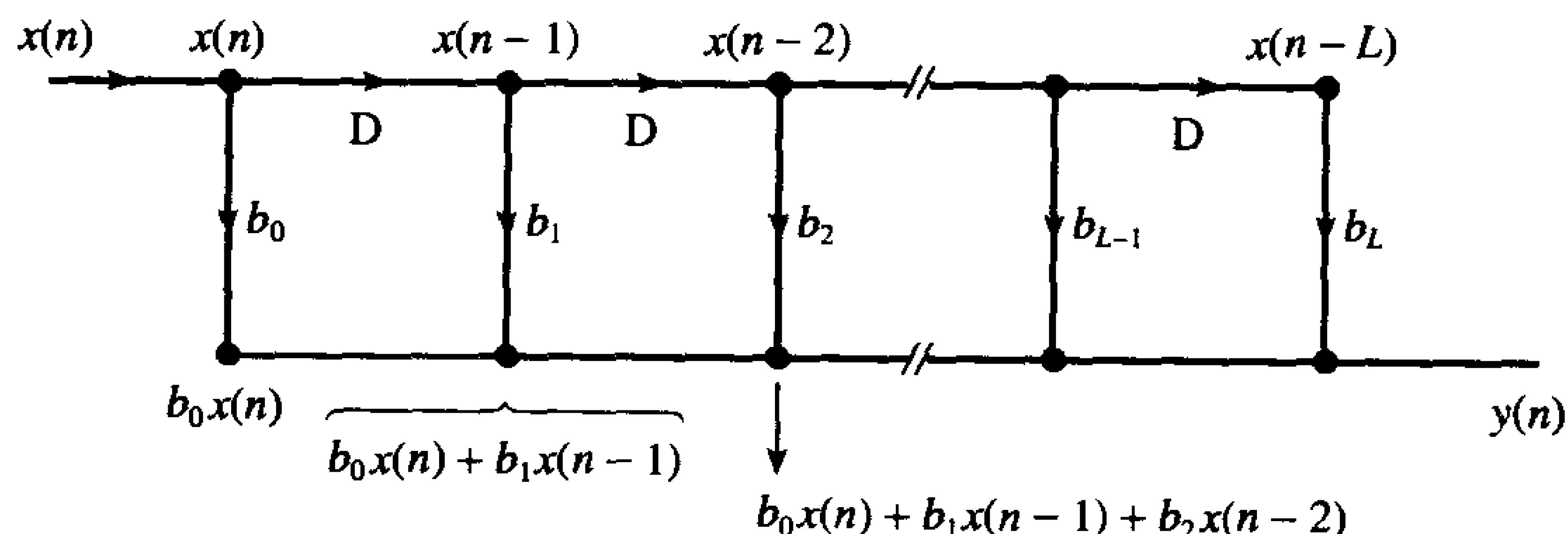


图5.38 一个非递归滤波器的框图表示

$$y(0) = h(0) = b_0x(0) + b_1 \times 0 = b_0 \times 1 = b_0$$
$$y(1) = h(1) = b_0 \times 0 + b_1x(0) + b_2 \times 0 = b_1 \times 1 = b_1$$
$$\vdots$$
$$y(L) = h(L) = b_0 \times 0 + 0 + 0 + \ldots + 0 + b_L \times 1 = b_L$$

因此

$$h(n) = \{b_0, b_1, \ldots, b_L\} \tag{5.128}$$

这说明了系统框图上的权值刚好对应于它的冲激响应函数的系数。这样的系统我们称之为有限冲激响应（FIR）滤波器。

现在考虑相对应于一般输入序列 $x(n)$的输出。把连续值代入到 5.127 式中，得

$$\begin{aligned} y(n) &= b_0x(n) + b_1x(n-1) + \ldots + b_nx(0) \\ &\equiv h(0)x(n) + h(1)x(n-1) + \ldots + h(n)x(0) \end{aligned} \tag{5.129}$$

这可以认为是输入和输出的卷积，如我们期望的那样。因此FIR滤波器也可以看作为卷积器，其中滤波器的权值对应于它们的冲激响应的系数。

在无限冲激响应（IIR）滤波器情况下，有不同但类似的关系式。考虑一阶递归滤波器，它的描述如下：

$$y(n) = a_1y(n-1) + b_0x(n) \tag{5.130}$$

对于一个单位冲激输入，很容易证明：

$$y(n) = h(n) = b_0a_1^n \quad n \geqslant 0 \tag{5.131}$$

对于一般的输入序列 $x(n)$，假设 $y(-1) = 0$，

$$y(0) = b_0x(0)$$
$$y(1) = a_1b_0x(0) + b_0x(1)$$
$$y(2) = a_1^2b_0x(0) + a_1b_0x(1) + b_0x(2)$$
$$\vdots$$
$$y(n) = a_1^nb_0x(0) + a_1^{n-1}b_0x(1) + \ldots + a_1b_0x(n-1) + b_0x(n)$$

把已知的权值代入，从 5.131 式可得：

$$y(n) = h(n)x(0) + h(n-1)x(1) + \ldots + h(0)x(n) \tag{5.132}$$

5.131 式和 5.132 式说明了对应于一阶系统的 IIR 滤波器是一个卷积器，它的冲激响应系统 $h(n) = b_0a_1^n$。

FIR 滤波器可以用在语音处理中（Grant et al., 1989），用以达到减少 PCM 的带宽；也可以用于子带编码器、频谱分析；还可以用于线性预测声码器。FIR 滤波器还可以用在雷达、扩展频谱通信中（Grant et al., 1989）。

5.5.2.2　卷积编码

卷积码允许对突发误差进行校正，这是通过对一个长的符号流分配码位奇偶校验来实现的（Stremler, 1982; Taub and Schilling, 1986）。一个移位寄存器的触发输出提供延迟，并且被抽头抽取，

然后模2加法器将进行合适的组合。这样就产生了许多输出，这些输出被每一个时钟周期连续地读出（参见图5.39）。这个系统本质上是因果的、非递归的，产生的输出依赖于以前的输入，并把新的输入数据和它的冲激响应进行卷积。

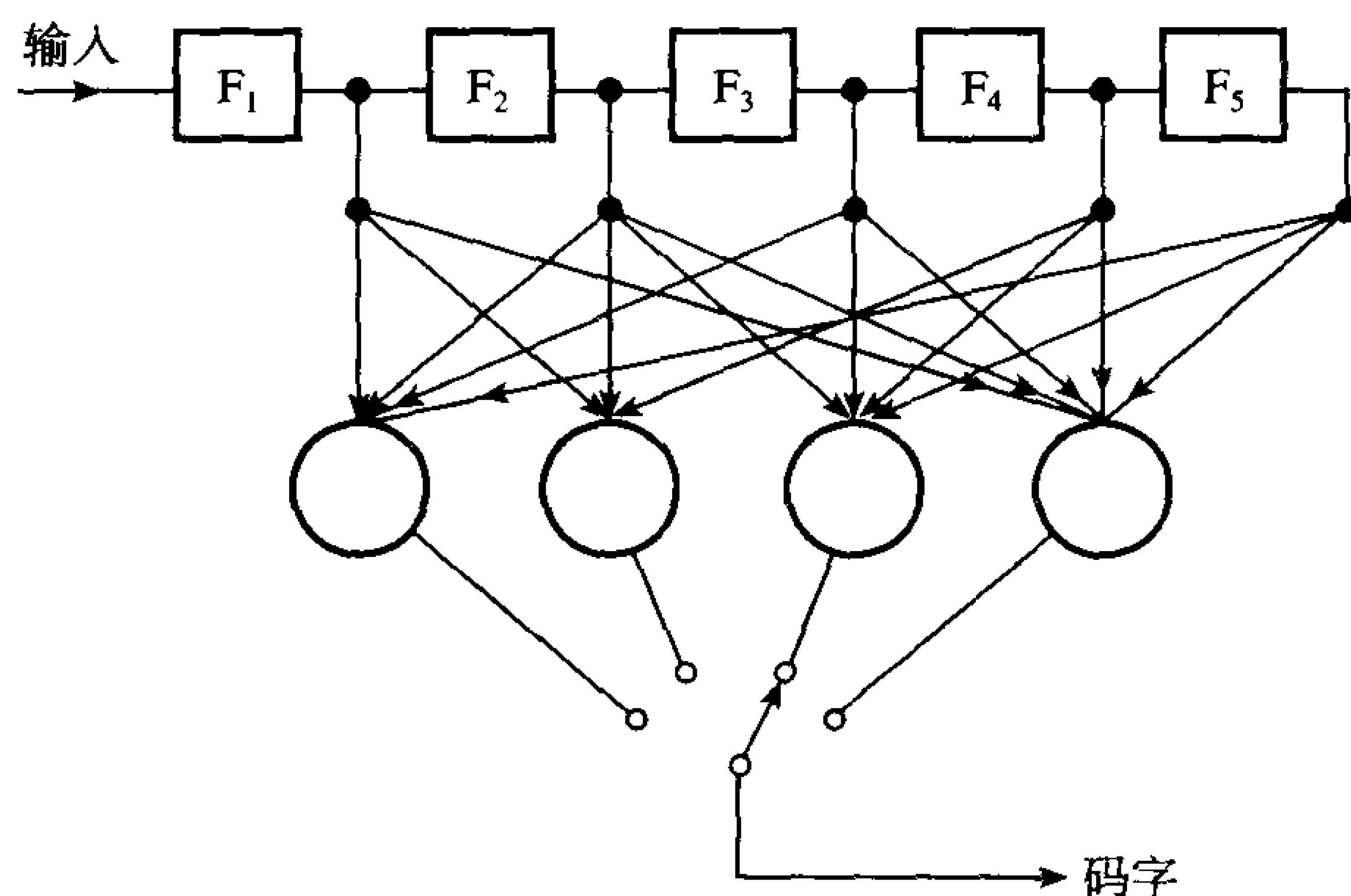


图5.39 卷积编码器

5.5.2.3 反卷积

如果系统的所有输入和系统的冲激响应进行卷积，可能会使输出失真。例如这种情况可能发生在通信系统中，对此必须设计一个均衡器，这个均衡器是一个线性滤波器，能对卷积的输出反卷积。在设计一个反卷积滤波器之前，必须测量系统的冲激响应（系统识别）。关于系统识别和反卷积的主题相当广泛（Proakis and Manolakis, 1988），这里我们不做讨论。

5.5.2.4 语音

我们对语音分析和编码有很大的兴趣，因为它可以用于人机交互和数据压缩。语音应用利用了这样的事实，语音波形可以看作为一系列表示音的脉冲、激励脉冲和声道的冲激响应的卷积（Rabiner and Gold，1975）。最后得出的三卷积可以很容易地转换成一个适合LTI系统处理的形式。FIR滤波器在语音处理中的应用在5.5.2.1节中着重进行了介绍。

5.6 小结

相关和卷积这个主题以及它们的相互关系在本章中进行了深入的讨论，同时也对相关的标准过程和避免尾端效应进行了讨论。本章还讨论了对含噪声信号的相关、通过相关方法识别噪声中的信号以及其他一些应用。我们还对基于相关和卷积定理并利用FFT的快速相关和卷积技术进行了描述，并说明了如何得到线性卷积。为了得到一个长数据序列的卷积，我们采用快速重叠相加和重叠保留技术。本章还描述了利用FFT的实部和虚部以将实数运算的速度提高2倍。

习题

5.1 两个不同的等长度的记录由一个周期性的脉冲列组成，它们沿着一个含噪声的信道发射。表5.4给出了抽样电压的记录值。

(1) 求两个记录之间延迟的数目，以及波形的周期。

(2) 推导周期波形。

表 5.4　从同一个信道得到的两个不同记录的抽样电压（伏特）

记录 1	6.02	−5.98	7.92	−7.96	−0.78
记录 2	8.93	−7.20	−0.82	3.23	1.44
	−8.34	9.22	−2.65	−3.7	9.51
	5.43	−9.88	−1.13	0.79	9.83
	5.53	3.50	−3.18	−8.85	8.21
	−8.73	4.64	−8.49	−4.66	−8.84
	1.69	−0.06	6.65	−8.00	−9.21
	5.55	−8.24	−0.37	2.71	4.63
	−0.78	7.27	−5.98	−3.97	9.11
	1.88	−0.92	−5.33	9.01	9.23
	4.23	2.99	−1.85	−5.27	3.81
	−3.7	5.08	−0.72	−5.08	−2.6
	6.62	−2.64	2.08	−5.91	−3.58
	9.67	−8.55	−3.08	4.18	8.11
	−1.65	3.64	−8.19	−3.50	4.84
	0.74	−3.87	−4.09	8.03	6.91
	7.25	2.93	−4.42	−8.21	3.61
	−9.87	−3.62	−8.29	−5.8	−7.04

5.2　在有尾端效应校正和无校正的两种情况下，计算记录 1 和 2 的互相关函数。估计由尾端效应引起的误差。

5.3　记录 1 和记录 2 在零延迟时，计算得到的相关百分比是多少？假设相关百分比定义为相关系数 ρ_{12} 乘以 100%。

5.4　表 5.5 给出了一个含噪声波形的抽样电压。利用所求波形和样板波形的互相关技术来求出现的周期性波形的确切形状。并利用另一种方法来检查你的结论。

表 5.5　一个含噪声波形的抽样电压

−7.37,	−7.99,	3.31,	−8.59,	−1.68,	3.01,	12.21,	−2.38,	7.46,
−9.84,	1.48,	1.1、	−1.8,	5.48,	8.93,	0,	−9.36,	−10.11,
1.61、	3.36,	−4.86,	6.27					

5.5　计算习题 5.4 里周期波形的自相关函数。(a) 用数值的方法，(b) 解析方法。互相比较这些结果，并和含噪声波形的自相关函数相比。对它和预期的结果的差异进行说明。

5.6　对一个电压波形进行抽样和数字化。数字化的电压值如表 5.6 所示。确定波形是否可以看成随机的。假设抽样间隔是 1 ms，并且出现周期性信号分量，周期为 4 ms，求周期性波形的估计并画出它。

表 5.6　数字化的电压值

−0.92,	−3.71,	3.11,	−0.24,	4.65,	0.84,	−2.98,	−3.94,	−4.03,	−2.51,	0.17,
3.85,	2.58,	0.38,	4.58,	3.4,	−3.46					

5.7　比较下面情形的信噪比，

(1) 表 5.4 中记录 1 的含噪声的周期波形；

(2) 表 5.4 中记录 1 的自相关函数；

(3) 表 5.4 中记录 1 和 2 的互相关函数。

5.8　对下面情形计算其理论上的信噪比，

(1) 利用合适的脉冲列，从表 5.4 中记录 1 的数据通过互相关得到周期性波形；

(2) 表5.4中记录1的自相关函数，给定(i) 含噪声的正弦信号的自相关函数的信噪比$(S/N)_{r0}$是

$$(S/N)_{r0} = \frac{N}{1 + 8\Big/\left(\frac{S_i}{N_i}\right) + 2\Big/\left(\frac{S_i}{N_i}\right)^2}$$

其中N是数据的数目，S_i是信号功率，N_i是噪声功率；

(ii) 含噪声的正弦信号和一个具有同样周期的周期性脉冲列的互相关函数的信噪比$(S/N)_\delta$是

$$(S/N)_\delta = \frac{N}{1 + 1\Big/\left(\frac{S_i}{N_i}\right)}$$

5.9　比较习题5.7和习题5.8得到的答案。

5.10　一个匹配滤波器具有冲激响应函数$h(n) = \{1, -1, -1, 1, 1, -1, 1\}$，用来检测沿信道到达接收机的对应于发射的信号。表5.7给出了抽样信号值，它们每一个都表示一个幅度为±1.5 V、脉冲宽带为1 μs的双极性脉冲序列中的一个脉冲值。求信号的到达时间和匹配滤波器的常数值。

表5.7　从一个噪声双极信号中取得的抽样电压

t (μs)	0	1	2	3	4	5	6
电压	0.14	0.48	1.61	2.09	−2.40	0.40	2.35
t (μs)	7	8	9	10	11	12	13
电压	−0.59	−1.81	0.32	−0.47	1.81	−1.63	−2.28

5.11　求一个系统的冲激响应函数。该系统对PN序列$\{1, 1, -1, 1, -1, -1, 1, -1\}$的响应是$y(n) = \{0, 0, 0.5, 1.5, 1.5, 1.5, 1, -1, -1, -1, -1.5, -0.5, -0.5, -0.5\}$。

5.12　一个宽度为2 ms的单位幅度脉冲加到一个电路，该电路的冲激响应如图5.40所示。用数值的方法确定输出波形。以0.5 ms的间隔对波形抽样。

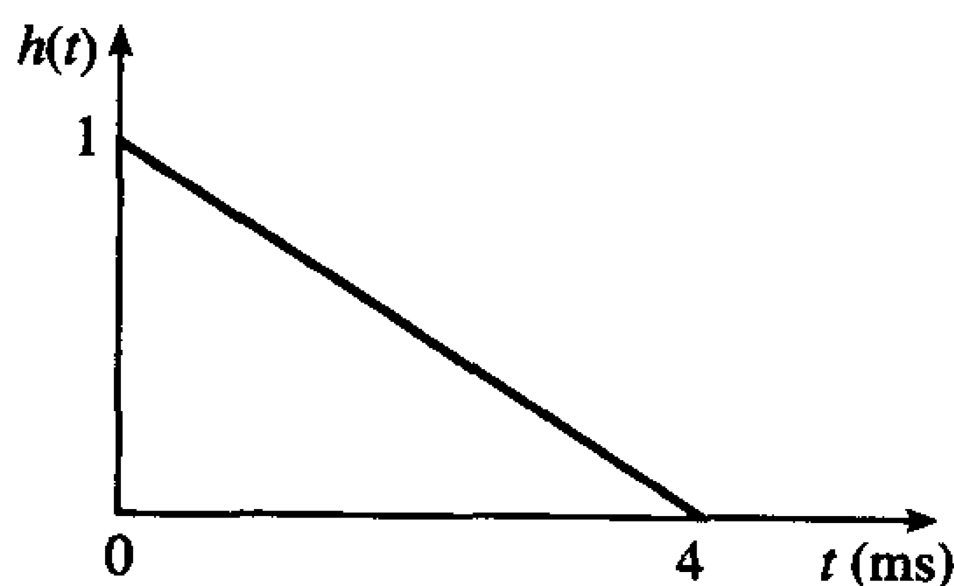


图5.40　习题5.12的系统冲激响应

5.13 (1) 图5.41给出了两个函数$x_1(t)$和$x_2(t)$，

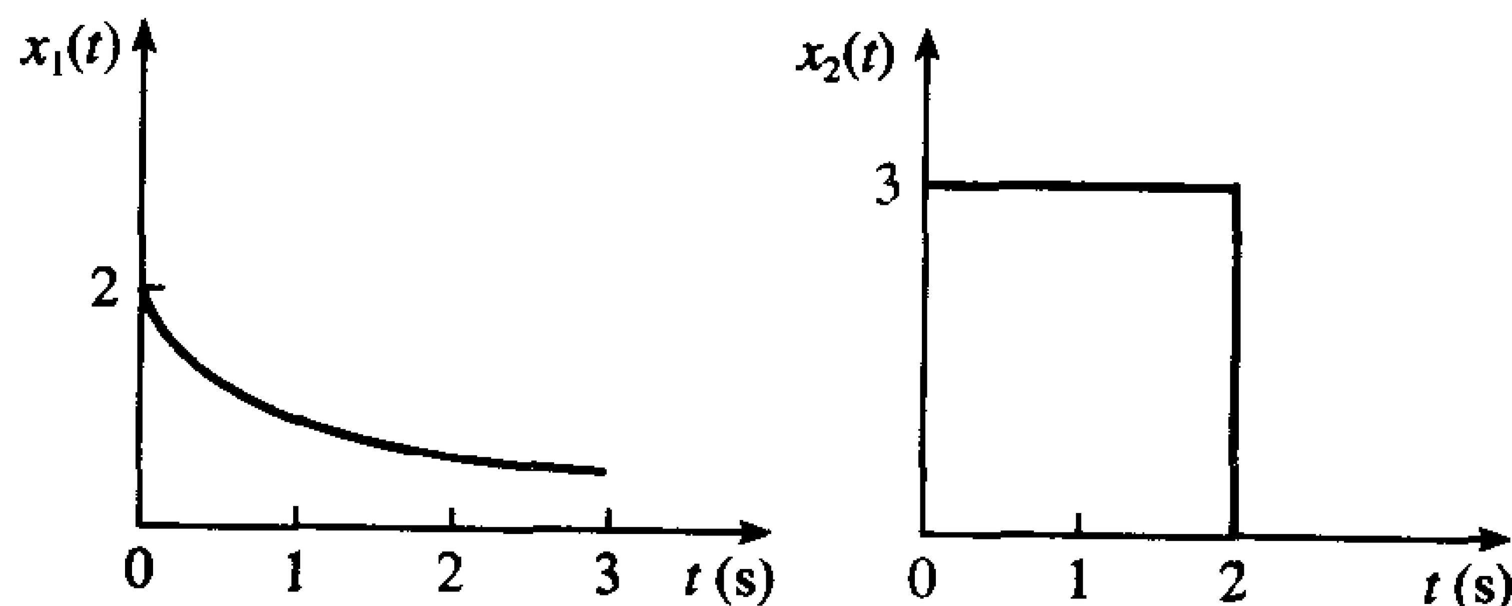


图5.41　习题5.13里的函数$x_1(t)$和$x_2(t)$

(a) 用数值的方法计算它们的卷积 $x_3(t)$，在 $t = 0, 1, 2, 3, 4, 5$ s 处取抽样值；

(b) 用解析的方法求 $x_3(t)$。

(2) 画出函数 $x_3(t)$，并对它们之间的不同给出解释。

5.14 当一个幅度为 5 V、宽度是 0.4 μs 的矩形脉冲，输入给一个截止频率是 6 MHz 的单级低通 RC 滤波器时，求输出脉冲的形状。假设滤波器的冲激响应如下：

$$h(t) = \frac{1}{CR} e^{-t/CR} u(t)$$

5.15 一个高度为 5 V、宽度为 1.0 μs 的矩形脉冲，输入到一个响应函数 $h(t)$如下所示的系统：

$$\begin{aligned} h(t) &= 0.1[1 - e^{-t/(1.09\times10^{-6})}] & 0 \leqslant t \leqslant 10\ \mu s \\ &= 0 & 10\ \mu s < t < 0 \end{aligned}$$

求系统的输出，

(1) 解析地求解；

(2) 每 1 μs 对 $h(t)$抽样，用一个位于 $t = 0$ s 的冲激函数来表示该脉冲。

比较你的结果。

5.16 求两个数据序列{1.5, 2.0, 1.5, 2.0, 2.5}和{0, 0.33, 0.67, 1.0}的互相关函数，

(1) 用直接互相关法；

(2) 用相关定理。

5.17 当加入的输入是{0, 2.5, 5.0, 0}（V）时，求一个冲激响应函数为{0, 0.899, 0.990, 0.991, 1}的电子系统的输出，

(1) 用直接互相关法；

(2) 用卷积定理。

5.18 利用重叠相加法计算系统的输出，该系统的冲激响应函数 $h(n)$ = {0, 0.899, 0.990, 0.999,1}，输入数据如表 5.5 所示（但是忽略了最后两个数据）。假设数据是以 2.5 μs 的间隔来抽样的，输入数据分成相等的五段。计算系统输入和输出之间的相位偏移。利用直接卷积法检验你的结果。

5.19 利用重叠相加法重做习题 5.18，其中卷积是用卷积定理求，并把结果和习题 5.18 的结果相比较。

5.20 利用重叠保留法求习题5.18中系统的输出，系统的输入数据为表5.5中除了最后两个数据之外的所有数据，冲激响应函数 $h(n)$ = {0, 0.899, 0.990, 0.999, 1}。把结果和习题 5.18 的结果相比较。

5.21 应用卷积定理来计算卷积，重做习题 5.20。把该结果和习题 5.18 ~ 习题 5.20 相比较。

5.22 考虑习题 5.18 ~ 习题 5.21 以及自己的解，相互比较不同方法所要求的计算量，并把它们和直接卷积要求的计算量相比较。

5.23 编写一个用重叠相加法来执行卷积的程序。用它来确认习题 5.18 的结果，然后考察不同的系统对你选择的输入的输出。

5.24 (1) 编写一个程序来执行快速相关，用它来求表 5.4 中记录 1 和记录 2 的互相关。

(2) 考察不同波形例如正方形波、矩形波、正弦波、随机噪声和不同信噪比波形的自相关和互相关。

(3) 比较用相关和频谱估计法来检测噪声中信号的能力。

MATLAB 习题

5.25 两个离散时间信号的连续抽样值如下：

$$x = 4, 2, -1, 3, -2, -6, -5, 4, 5$$
$$y = -4, 1, 3, 7, 4, -2, -8, -2, 1$$

(a) 利用 MATLAB，计算每个数据序列归一化和非归一化的自相关函数的估计。

(b) 利用 MATLAB，计算并画出每个数据序列有偏和无偏自相关函数的估计。

(c) 利用 MATLAB，计算并画出两个数据序列归一化和非归一化的自相关函数的估计。

(d) 利用 MATLAB，计算并画出两个数据序列归一化和非归一化的有偏和无偏自相关函数的估计。

(e) 对上面的(a)和(b)，求其在零延时的互相关或自相关函数的估计。

(f) 对上面的(a)和(b)，求互相关或者自相关函数的长度。

(g) 比较上面(a)和(b)的结果，并对它们之间的差异进行评论。

5.26 对习题 5.25(a)中得到的数据序列，求其归一化的互相关函数，并交换数据序列，求其归一化的互相关函数（即 x 和 y 的互相关与 y 和 x 的互相关）。

5.27 (a) 产生一个 1000 点的随机高斯白噪声数据序列（利用函数 randn）。

(b) 对于前 30 个延迟，计算并画出(a)中的序列的自相关函数的估计。

5.28 一个连续时间信号特性由下面的等式规定：

$$x(t) = A\cos(2\pi f_1 t) + B\cos(2\pi f_2 t)$$

(a) 在 MATLAB 的协助下，产生一个和信号等价的离散时间序列。假设抽样频率是 1 kHz，$f_1 = 50$ Hz，$f_2 = 100$ Hz，两个频率分量的幅度之比 $A/B = 1.5$。

(b) 计算并画出(a)中序列的自相关函数的估计。

5.29 (a) 在适当的 MATLAB 函数的协助下，产生以下的波形，并画出它们：

(i) 一个正弦波——利用 sin(2*pi*t/100)，$t = 0:1:1000$。

(ii) 一个噪声波形——利用函数 randn。

(iii) 一个含噪声的正弦波形——把波形(i)和(ii)相加。

(iv) 一个矩形形波——利用 *square*(2**pi***t*/100)。

(b) 对(a)里的每一个波形，计算并画出其归一化的自相关函数。

(c) 简要描述(b)里计算出来的自相关函数独有的特性和共性。

5.30 在这个习题中，我们的目标是模拟一个通过相关来估计目标距离的问题。发射信号和从目标反射回来的含噪声信号之间的互相关函数显现出一个峰值，该峰值所在的时间延迟对应于要求的距离的 2 倍。

(a) 在合适的 MATLAB 函数的协助下，产生图 1.3(a)和图 1.3(b)描绘出的波形（其中上面和下面的图形分别代表发射和接收的波形）。

(b) 计算两个波形之间的互相关函数，并估计从发射机到目标之间的距离。

假设无线电波传输速度为 3×10^8 m/s，抽样频率是 4 MHz。

5.31 利用 xcov 函数，重复习题 5.25。把你得到的结论和从习题 5.25 得到的相比较，并评述其差异。

参考文献

Beauchamp K.G. (1973) *Signal Processing Using Analog and Digital Techniques*. London: Allen and Unwin.

Bell A.J. and Sejnowski T.J. (1995) An information-maximisation approach to blind separation and blind deconvolution. *Neural Computation*, **7**, 1129–59.

Brigham E.O. (1974) *The Fast Fourier Transform*, Sections 13.3 and 13.4. Englewood Cliffs NJ: Prentice-Hall.

Chatfield C. (1980) *The Analysis of Time Series*, p. 62. London: Chapman and Hall.

DeFatta D.J., Lucas J.G. and Hodgkiss W.S. (1988) *Digital Signal Processing: A System Design Approach*, Section 6.9, p. 306. New York: Wiley.

Grant P.M., Cowan C.F.N., Mulgrew B. and Dripps J.H. (1989) *Analogue and Digital Signal Processing and Coding*, Chapters 16, 17, 19 and 20. Bromley, UK: Chartwell-Bratt.

Jenkins G.M. and Watts D.G. (1968) *Spectral Analysis and its Applications*. San Francisco CA: Holden-Day.

Jervis B.W., Goude A., Thomlinson M., Mir S. and Miller G. (1990) Least squares artefact removal by transputer. In *IEE Colloquium on the Transputer and Signal Processing*, Savoy Place, London, 5 March 1990.

Main G. and Howell T.D. (1993) Determining a signal to noise ratio for an arbitrary data sequence by a time domain analysis. *IEEE Transactions on Magnetics*, **29**(6), November, 3999–4001.

McGillem C.D. and Cooper G.R. (1974) *Continuous and Discrete Signal and System Analysis*. New York: Holt, Rinehart, and Winston.

Proakis J.G. and Manolakis D.G. (1988) *Introduction to Digital Signal Processing*, p. 429. Basingstoke: Macmillan.

Rabiner L.R. and Gold B. (1975) *Theory and Application of Digital Signal Processing*, Chapters 12 and 13. Englewood Cliffs NJ: Prentice-Hall.

Stremler F.G. (1982) *Introduction to Communication Systems*, 2nd edn, Section 3.10 and p. 407. Reading MA: Addison-Wesley.

Strum R.D. and Kirk D.E. (1988) *First Principles of Discrete Systems and Digital Signal Processing*, Chapter 3. Reading MA: Addison-Wesley.

Taub H. and Schilling D.L. (1986) *Principles of Communication Systems*, 2nd edn, p. 562. New York: McGraw-Hill.

附录

5A 计算互相关和自相关的C语言程序

计算互相关和自相关的C语言程序correltn.c在指导手册的CD上可以找到（详情请参见前言）。

第6章　数字滤波器的设计框架

这一章的目的是为数字滤波器的设计提供一个通用的框架。对于数字滤波器的设计，本章描述了一个简单的、从技术规范到实现的逐步设计指南。在设计过程的每个步骤设计者所需要面临的选项以及影响对这些选项进行选择的因素，我们利用了几个说明性的例子对它们进行了重点强调。大多数的DSP教材对于数字滤波器理论都提供了许多篇幅，尤其是对近似方式，这反映在寻找计算滤波器系数的有用方法方面已经做了大量的研究工作，并且在滤波器设计方面已经取得了显著的进展。然而，这些内容会使一个没有经验的滤波器设计者感到无所适从，他或者她实际上不知道究竟如何去设计一个滤波器，以及怎么把这个滤波器组合起来。因此，以我们的经验来说，对于那些不是从纯理论观点而是想实际设计数字滤波器的设计者来说，本章提供的这个框架是非常有价值的。本章为第7章和第8章做好了铺垫，那两章涵盖了实际数字滤波器的设计。

6.1　数字滤波器概述

一个滤波器实质上是一个系统或者网络，它以一种期望的模式有选择地改变信号的波形、幅度-频率和/或相位-频率特性。一般滤波的目的是为了改善一个信号的质量（例如消除或者减少噪声），或者从信号中提取信息，或者是把以前为了有效地利用通信信道而组合在一起的两个或多个信号分离出来。

如同我们后面将看到的那样，数字滤波器是用硬件或者软件实现的一种算法，这个算法是为了达到滤波的目的而对数字输入信号进行运算产生数字输出信号。数字滤波器这个词是指执行滤波算法的特定硬件或者软件程序。数字滤波器经常作用的对象是数字化的模拟信号，或者刚好是存储在计算机存储器里代表某些变量的数。

图6.1给出了一个具有模拟输入信号和输出信号的实时数字滤波器的简化框图。这个带限模拟信号被周期地抽样，且转化成一系列数字$x(n)$（$n = 0,1,...$）。数字处理器依据滤波器的计算算法，执行滤波运算、把输入系列$x(n)$映射到输出系列$y(n)$。DAC把数字滤波后的输出转化成模拟值，这些模拟值接着被模拟滤波器平滑，并且消去不想要的高频分量。

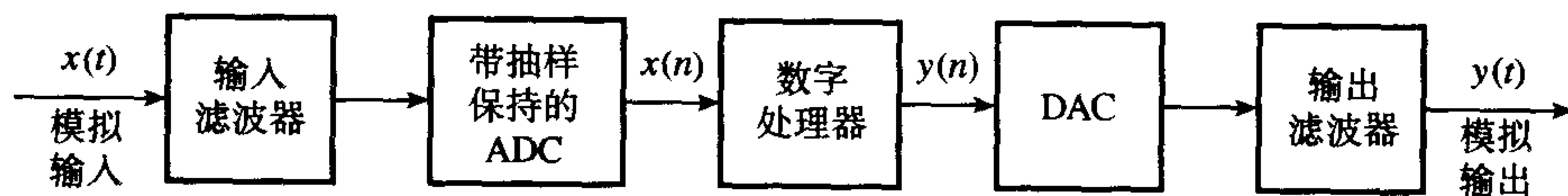


图6.1　一个具有模拟输入和输出信号的实时数字滤波器的简化框图

数字滤波器在DSP里具有非常重要的作用。在许多应用中（例如数据压缩，生物医学信号处理，语音处理，图像处理，数据传输，数字音频，电话回声对消，等等），数字滤波器和模拟滤波器相比因为具有如下一个或多个优势而被优先采用。

- 数字滤波器可以具有模拟滤波器不可能有的某些特性，例如真正的线性相位响应。
- 数字滤波器的性能不像模拟滤波器那样随环境的改变（例如温度的变化）而改变。这样就不必经常去校正。

- 如果利用一个可编程的处理器来实现，那么数字滤波器的频率响应能被自动地调整。这就是为什么在自适应滤波器中广泛利用数字滤波器的原因。
- 几个输入信号或通道可以用一个数字滤波器来滤波，而不需要重复硬件。
- 滤波过的和未滤波的数据都可以将其存储以备将来使用。
- 可以充分利用在VLSI技术方面的技术进展来制造数字滤波器，使滤波器体积更小、功耗低、价格便宜。
- 在实践中，模拟滤波器能达到的精度是受限制的。例如，利用现有的元件设计的有源滤波器，通常可能达到的最大阻带衰减是60 ~ 70 dB。而对于数字滤波器，它的精度仅受限于它采用的字长。
- 数字滤波器的性能从单元到单元是可以重复的。
- 数字滤波器可以以极低的频率运行，例如在生物医学中有许多极低频率应用的例子，在这些应用中采用模拟滤波器是不现实的。另外，数字滤波器仅通过改变抽样频率就可以在很大的频率范围内工作。

和模拟滤波器相比，数字滤波器主要有以下缺点：

- **速度限制** 数字滤波器能实时处理的最大信号带宽，比模拟滤波器低得多。在实时情况下，模拟-数字-模拟转化过程对数字滤波器的性能引入了速度的限制。ADC的转换时间和DAC的建立时间限制了能够处理的最高频率。此外，数字滤波器的运行速度，依赖于所用到的数字处理器的速度，以及滤波算法必须执行的算术操作的数目。滤波器的响应越紧凑，则滤波器的速度越快。
- **有限字长效应** 数字滤波器受由于量化一个连续信号而引起的ADC噪声的影响，以及在计算过程中发生的舍入噪声的影响。递归滤波器的阶数越高，舍入噪声的累计就越大，可能会引起滤波器的不稳定。
- **设计和开发期限长** 数字滤波器的设计和开发期限，特别是硬件的开发可能比模拟滤波器的长得多。不过，一旦硬件和/或软件开发出来，不需要或者稍加变动就可以将其用在别的滤波任务或者DSP任务中（在随后的章节中给出了一些例子）。好的计算机辅助设计（CAD）支持软件使得设计滤波器成为一项令人愉快的任务，但是如何充分而有效地利用这些辅助工具就需要专门的技术了。

6.2 数字滤波器的类型：FIR和IIR滤波器

数字滤波器分为两大类，即无限冲激响应（IIR）和有限冲激响应（FIR）滤波器。在基本形式上，每一种滤波器都可以用它的冲激响应序列 $h(k)$（$k = 0, 1,\ldots$）来表示，如图6.2所示。滤波器的输入和输出信号通过卷积和相联系，6.1式给出了IIR滤波器的相关公式，6.2式给出了FIR滤波器的相关公式。

$$y(n) = \sum_{k=0}^{\infty} h(k)x(n-k) \tag{6.1}$$

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k) \tag{6.2}$$

从这些等式可知，IIR滤波器的冲激响应具有无限的持续时间，而FIR滤波器的冲激响应具有有限持续时间，因为FIR的 $h(k)$ 只有 N 个值。在实际中，利用6.1式来计算IIR滤波器的输出是不可行的，因为它的冲激响应的长度太长（理论上是无穷大的）。IIR滤波器方程是用递归形式表示：

$$y(n)=\sum_{k=0}^{\infty}h(k)x(n-k)=\sum_{k=0}^{N}b_k x(n-k)-\sum_{k=1}^{M}a_k y(n-k) \tag{6.3}$$

其中a_k和b_k是滤波器的系数。因此，6.2式和6.3式分别是FIR和IIR滤波器的差分方程。这些方程，特别是FIR的$h(k)$的值，或者IIR的a_k和b_k的值，在大多数滤波器的设计问题中是非常重要的目标。我们注意到，在6.3式里，当前输出样本$y(n)$是过去输出以及现在和过去输入样本的函数，也就是IIR是一种反馈系统。这应该和FIR方程相比较，在FIR方程中，当前输出样本$y(n)$仅是过去和现在输入值的函数。然而，当令b_k为零时，6.3式化简为6.2式的FIR。

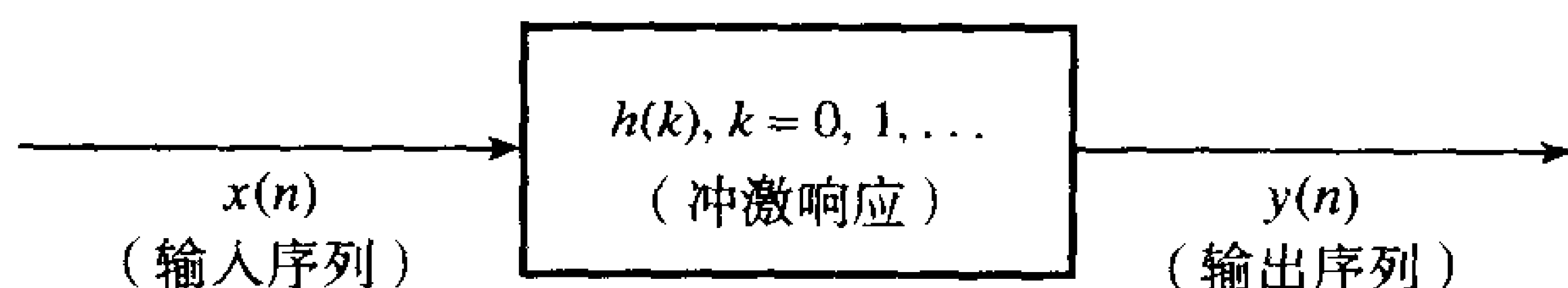


图6.2　数字滤波器的概念性的表示

6.4a式和6.4b式分别给出了FIR和IIR滤波器的另外一种表示。这两个式子是这些滤波器的传递函数，它们在评价滤波器的频率响应时是非常有用的（详情请参见第4章、第7章、第8章）。

在接下来新的一节里我们会很清晰地看到：一些因素滤波器设计者对选项进行选择，这些因素在数字滤波器设计过程的每一个阶段都呈现在设计者的面前。这些因素涉及到滤波器是IIR还是FIR的。因此，理解IIR和FIR的差别以及它们各自特有的性质是非常重要的。更为重要的是如何在它们之间进行选择。

$$H(z)=\sum_{k=0}^{N-1}h(k)z^{-k} \tag{6.4a}$$

$$H(z)=\sum_{k=0}^{N}b_k z^{-k}\Bigg/\left(1+\sum_{k=1}^{M}a_k z^{-k}\right) \tag{6.4b}$$

6.3　在FIR和IIR滤波器之间的选择

选择FIR或者IIR滤波器大体上依赖于两种滤波器的优点。

(1) FIR滤波器可以具有精确的线性相位响应。其潜在的含义就是采用这种滤波器不会给信号带来相位失真。这在许多应用中，例如数据传输、生物医学、数字音频和图像处理等，相位不失真是非常重要的要求。IIR滤波器的相位响应特别是在带沿为非线性的。

(2) FIR的实现是非递归的，也就是通过6.2式直接得出的结果，它总是稳定的。而IIR滤波器的稳定性不是一直都能得到保证的。

(3) 采用有限位数实现滤波器的影响，例如舍入噪声和系数量化误差，FIR比IIR要小得多。

(4) 对锐截止（sharp cutoff）滤波器，FIR要求的系数比IIR要多。因此对一个给定的幅度响应的规范，FIR实现要求更多的处理时间和更大的存储。然而，我们可以很容易地利用FFT的计算速度和多速率技术（参见第9章）来有效地提高FIR实现的效率。

(5) 模拟滤波器可以很容易地转化成等价的满足类似性能规范的IIR数字滤波器。使用FIR滤波器是不可能的，因为它没有对应的模拟滤波器。然而，通过FIR可以更容易地合成具有任意频率响应的滤波器。

(6) 一般来说，如果不提供CAD的支持，FIR的合成在数学上比IIR要较难得到。

综上所述，下面给出一个什么时候用FIR及什么时候用IIR的大致的指南。

- 当锐截止滤波器和高吞吐率是惟一重要的要求时，采用IIR滤波器。因为IIR滤波器，特别是具有椭圆特性的IIR滤波器，所需的系数比FIR少。
- 如果滤波器系数的数目不是太大，而且在实践中需要相位失真很小或者不能有相位失真，那么采用FIR滤波器。另外有一条要增加的是：新的DSP处理器具有适应于FIR滤波的结构，实际上某些DSP就具有针对FIR滤波的设计（参见第12章）。

例6.1 下列传递函数代表满足相同幅度–频率响应规范的两个不同的滤波器：

(1) $H(z) = \dfrac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$

其中

$$b_0 = 0.498\,181\,9$$
$$b_1 = 0.927\,477\,7$$
$$b_2 = 0.498\,181\,9$$
$$a_1 = -0.674\,487\,8$$
$$a_2 = -0.363\,348\,2$$

(2) $H(z) = \sum_{k=0}^{11} h(k) z^{-k}$

其中

$$h(0) = 0.546\,032\,80 \times 10^{-2} = h(11)$$
$$h(1) = -0.450\,687\,50 \times 10^{-1} = h(10)$$
$$h(2) = 0.691\,694\,20 \times 10^{-1} = h(9)$$
$$h(3) = -0.553\,843\,70 \times 10^{-1} = h(8)$$
$$h(4) = -0.634\,284\,10 \times 10^{-1} = h(7)$$
$$h(5) = 0.578\,924\,00 \times 10^{0} = h(6)$$

对于每一个滤波器，

(a) 说明它是FIR或者IIR滤波器；
(b) 用框图的形式表示滤波的运算，并写出差分方程；
(c) 确定并解释计算量和存储量。

解：

(a) 滤波器(1)和(2)分别是IIR和FIR。

(b) 滤波器(1)的框图如图6.3(a)所示，相应的差分方程为

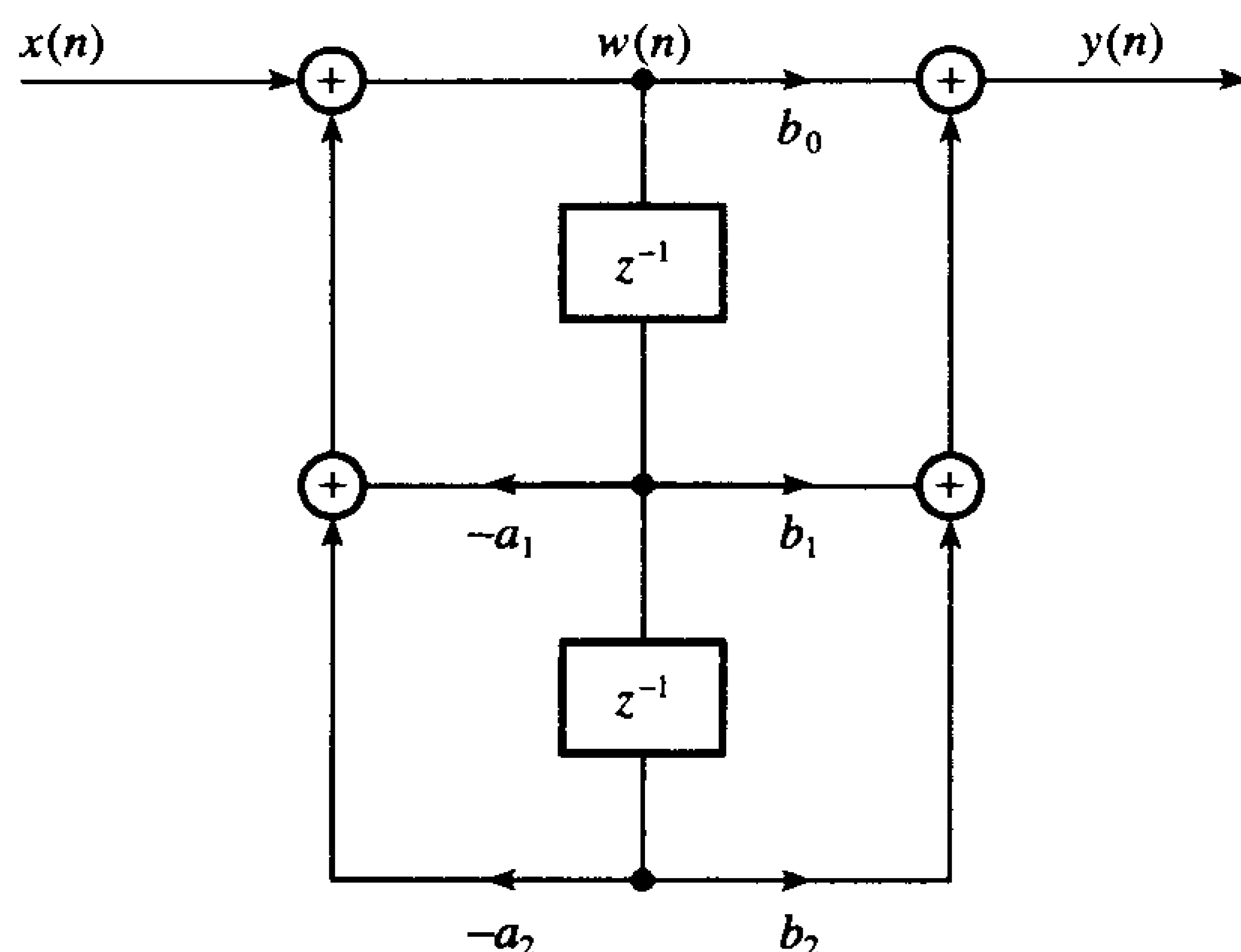
$$w(n) = x(n) - a_1 w(n-1) - a_2 w(n-2)$$
$$y(n) = b_0 w(n) + b_1 w(n-1) + b_2 w(n-2)$$

滤波器(2)的框图由图6.3(b)给出，相应的差分方程是为

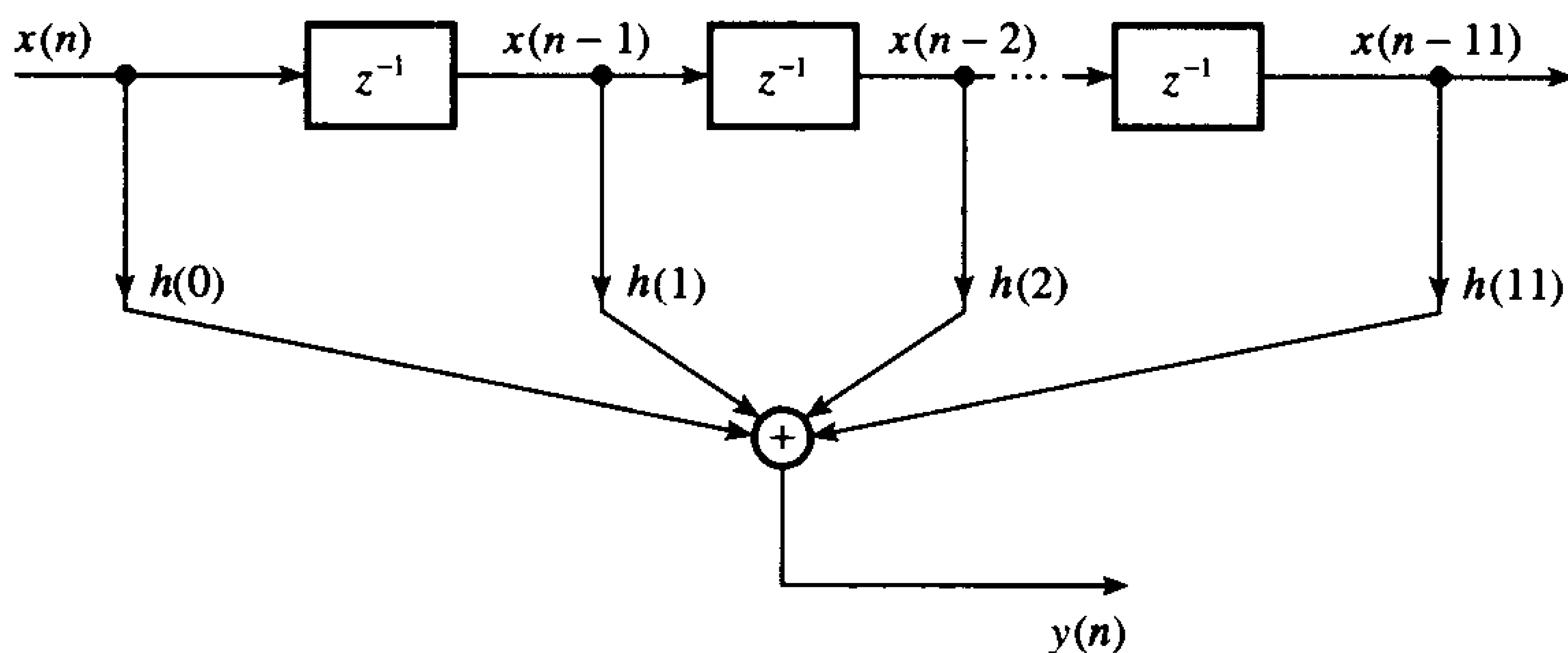
$$y(n) = \sum_{k=0}^{11} h(k) x(n-k)$$

(c) 通过检查两个差分方程，两个滤波器所需要的计算量和存储量可总结如下：

	FIR	IIR
乘法的数目	12	5
加法的数目	11	4
存储单元（系数和数据）	24	8



(a) 例6.1里的IIR滤波器的框图表示



(b) 例6.1里的FIR滤波器的框图表示

图6.3　例6.1里的IIR与FIR滤波器的框图表示

很显然IIR滤波器在计算和存储的要求上都比FIR滤波器经济。然而，我们可以利用FIR系数的对称性，使得FIR滤波器更有效，尽管很明显是以它实现的简单性为代价的。值得指出的一点是，对于同样的幅度响应规范，FIR滤波器的系数的数目（在这个例子里是12）通常是IIR传递函数的阶数（分母中z的最高幂，在本例中为2）的6倍。

6.4　滤波器的设计步骤

设计一个数字滤波器包括下面5个步骤：

(1) 滤波器要求的规范。

(2) 合适的滤波器系数的计算。

(3) 用一个适当的结构来表示滤波器（实现结构）。

(4) 有限字长效应对滤波器性能的影响的分析。

(5) 用软件和/或者硬件来实现滤波器。

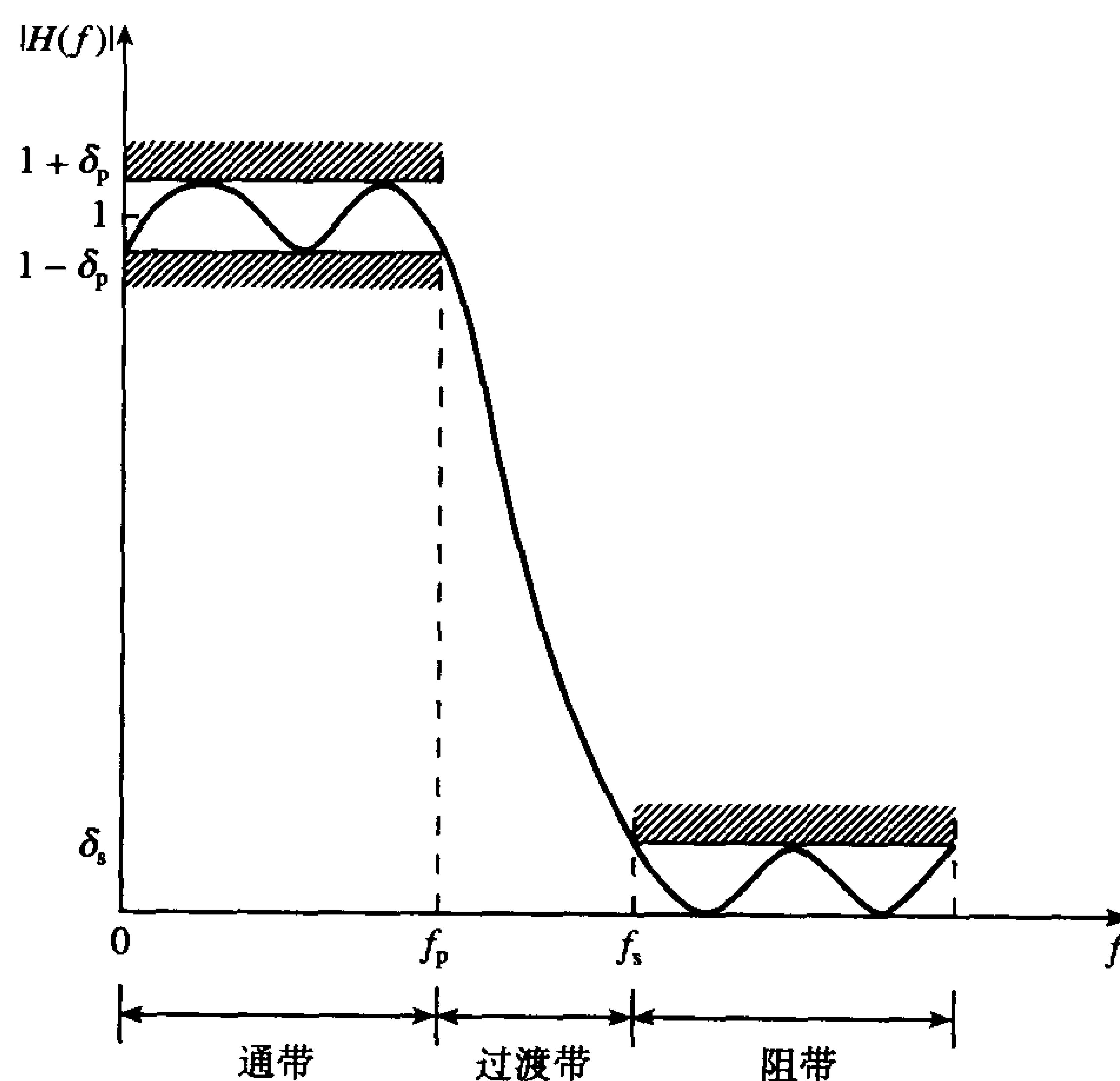
这五个步骤不是必须相互独立的，它们也不是总要按照上面给出的顺序执行。实际上把第二步、第三步和第四步组合在一起。不过，这里讨论的方法给出了一个简单的逐步指南将确保设计成

功。为了获得一个有效的滤波器，在这些步骤间重复几遍可能是必需的，特别是如果问题的规范本身就有漏洞（这是很常见的情况），或者如果设计者想探索其他可能的设计。接下来将详细地讨论这些步骤。

6.4.1 滤波器要求的规范

要求的性能规范包括：指定(i) 信号特性（信号源和接收器的类型，I/O 接口，数据率和带宽，感兴趣的最高频率），(ii) 滤波器的特性（期望的幅度和/或相位响应以及它们的容差（如果存在），滤波的执行速度和滤波模式（实时或者批处理））。(iii) 实现方法（例如，用计算机高级语言实现，或者用基于DSP处理器的系统实现，信号处理器的选择）。(iv) 其他的设计限制（例如，滤波器的成本）。设计者在一开始可能没有足够的信息来完全规定滤波器，但是尽可能多地指定滤波器的要求，这样可以简化设计过程。

尽管上面的要求依赖于应用，但在(ii)的某些方面花些时间将是有用的。数字滤波器的特性经常在频域定义。对于频率选择性的滤波器，例如低通和带通滤波器，性能规范经常以容差图的形式给出。图6.4给出了低通滤波器容差图，其中加了阴影的水平线表示容差的限度。在通带里，幅度响应有一个峰值偏差 δ_p；在阻带里，有一个最大偏差 δ_s。



6.4 一个低通滤波器的容差图

过渡带的宽度决定了滤波器的陡峭程度。在过渡带区域里，幅度响应从通带到阻带是单调下降的。下面是感兴趣的关键参数：

δ_p	通带偏差
δ_s	阻带偏差
f_p	通带边沿频率
f_s	阻带边沿频率

边沿频率经常以归一化形式给出，即以抽样频率（f/F_s）归一化，但是规范使用标准的频率单位赫兹或千赫是有用的，而且某些时候更有意义，特别是对于没有经验的设计者。当分别指定了通带波

纹和最小阻带衰减时，通带和阻带的偏差可以表示成普通的数字或者用分贝表示。因此最小阻带衰减A_s和峰值通带波纹A_p用分贝表示如下（对于 FIR 滤波器）：

$$A_s\text{（阻带衰减）} = -20\log_{10}\delta_s \quad (6.5a)$$

$$A_p\text{（通带波纹）} = 20\log_{10}(1+\delta_p) \quad (6.5b)$$

数字滤波器的相位响应并不像它的幅度响应那样精确规定。在许多情况下，指出要考虑相位失真或者需要得到线性相位响应就足够了。然而，在一些滤波器被用来均衡或者补偿一个系统相位响应的应用中，例如作为移相器，那么期望得到的相位响应将需要指定。

例 6.2　设计一个满足下面频率响应规范的 FIR 带通滤波器：

通带	0.18 ~ 0.33（归一化）
过渡带宽	0.04（归一化）
阻带偏差	0.001
通带偏差	0.05

(1) 画出滤波器的容差图。

(2) 用标准单位千赫表示出滤波器的通带边沿频率，假设抽样频率是 10 kHz，阻带和通带的偏差是用分贝表示。

解：

(1) 滤波器的容差表如图 6.5 所示。

(2) 在 10 kHz 的抽样频率下，通带边频、阻带和通带偏差给出如下：

通带	1.8 ~ 3.3 kHz
阻带	0 ~ 1.4 kHz 和 3.7 ~ 5 kHz
阻带衰减	$-20\log_{10}(0.001) = 60$ dB
通带波纹	$20\log_{10}(1+0.05) = 0.42$ dB

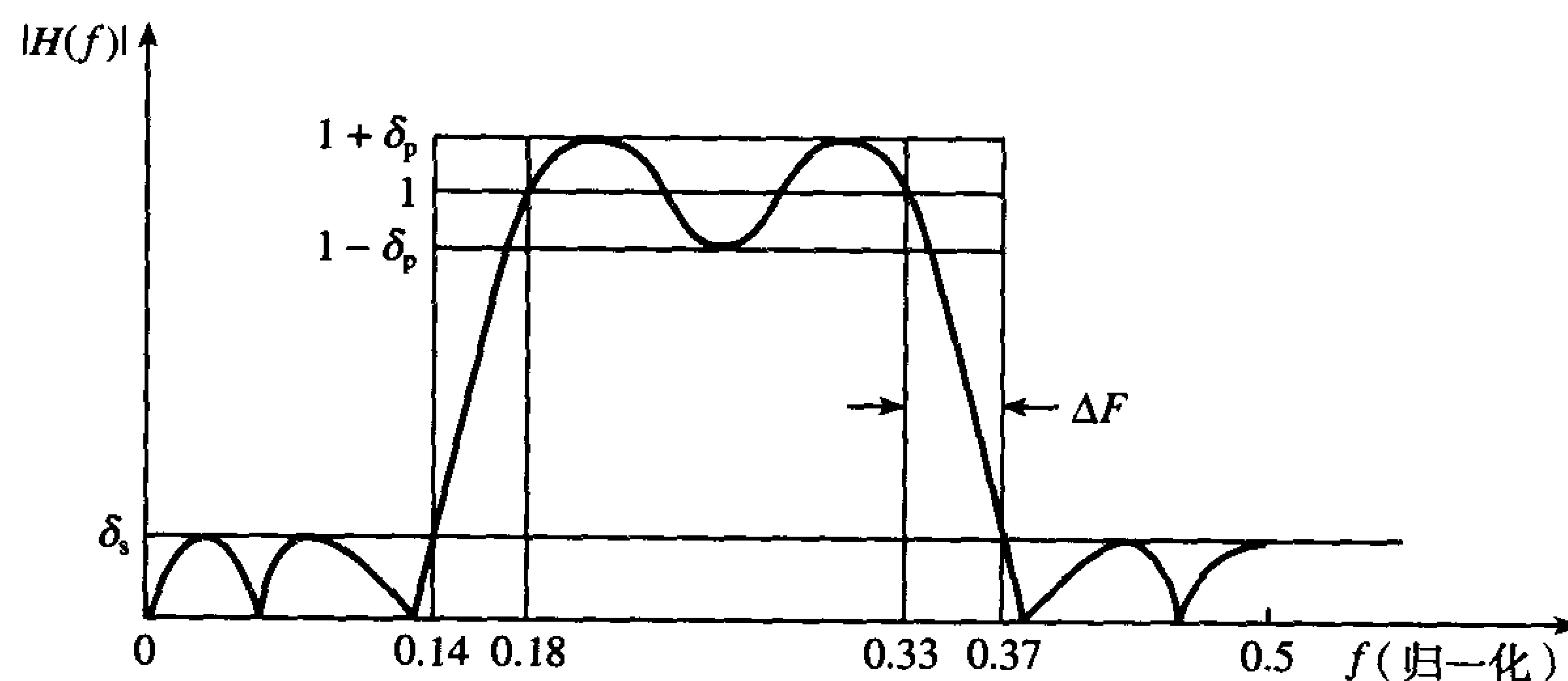


图 6.5　例 6.2 里的带通滤波器的容差图

6.4.2　系数计算

在这个步骤里，我们从一些近似方法中选出一种，以计算 FIR 的系数$h(k)$的值，或者 IIR 的a_k和b_k值，使得 6.4.1 节里给出的滤波器的特性得以满足。使用哪种方法来计算滤波器的系数依赖于滤波器的类型是 IIR 还是 FIR。

IIR滤波器系数的计算一般是根据已知的模拟滤波器的特性转换到等价的数字滤波器。两个常用的基本方法是冲激不变法和双线性变换法。在对模拟滤波器数字化以后，利用冲激不变法，原始的模拟滤波器的冲激响应得到保留，但不保留它的幅度-频率响应。因为固有的混叠，这个方法对高通或者带阻滤波器是不合适的。另一方面，双线性方法生成的滤波器非常有效，而且非常适合于频率选择型的滤波器系数的计算。它允许利用已知的典型特性例如巴特沃斯、切比雪夫（Chebyshev）和椭圆来设计数字滤波器。从双线性变换法得到的数字滤波器，一般来说保留了模拟滤波器的幅度响应特性，但时域特性没有保留。利用双线性法来计算滤波器系数的有效计算程序已经出现，它只需要指定感兴趣的滤波器系数（参见第8章）。冲激不变法对仿真模拟系统是很好的，但双线性法对频率选择的IIR滤波器是最好的。

极点-零点放置法提供了另一种计算IIR滤波器系数的方法。对于非常简单的滤波器来说，这种方法计算系数非常容易。不过，对于良好幅度响应的滤波器，不推荐使用点-零点放置法，因为它依赖于逐次逼近来逐渐移动极点和零点的位置。

和IIR滤波器一样，FIR滤波器也有计算系数的一些方法。在本书中详细讨论的三种方法是加窗、频率抽样和最优化（Parks-McClellan算法）。加窗法提供了一个简单、灵活的计算FIR滤波器系数的方法，但是它不能使设计者充分地控制滤波器的参数。频率抽样法的主要吸引力在于允许通过递归来实现FIR滤波器，FIR滤波器可以很高效地计算出来。不过，它在定义和控制滤波器参数上缺乏灵活度。利用高效、易用的程序的实用性，最优化法在工业上得到了广泛应用，对于大多数应用，它将产生期望的FIR滤波器。因此，对于FIR滤波器，最优化法应该是首选的方法，除非是特殊的应用暗示了使用其他的方法或者是无法利用CAD工具。

总之，有许多计算滤波器系数的方法，其中下面几种是最广泛使用的：

- 冲激不变法 （IIR）
- 双线性变换 （IIR）
- 极点-零点放置法 （IIR）
- 窗口方法 （FIR）
- 频率抽样 （FIR）
- 最优化 （FIR）

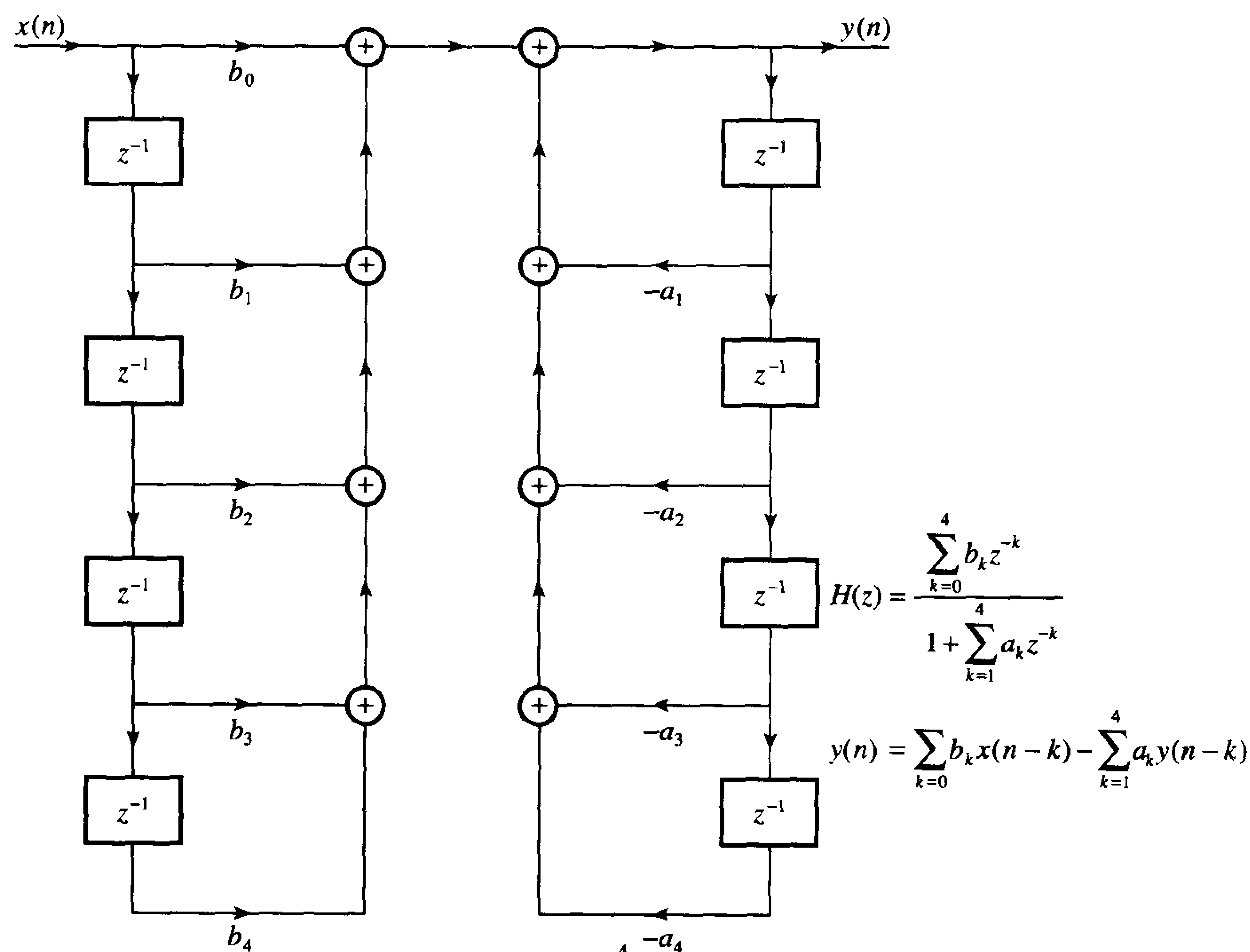
我们选择最适合实际应用的方法。我们的选择将受到几个因素的影响，其中最重要的是性能规范里边界频率的要求。一般来说，最关键的是在FIR与IIR之间的选择。在大多数情况下，如果FIR的性质是至关重要的，那么最优化法是较好的选择。同样，如果想要IIR特性，那么在大多数情况下，双线性法就足够了。

6.4.3 用适当的结构（实现）来表示滤波器

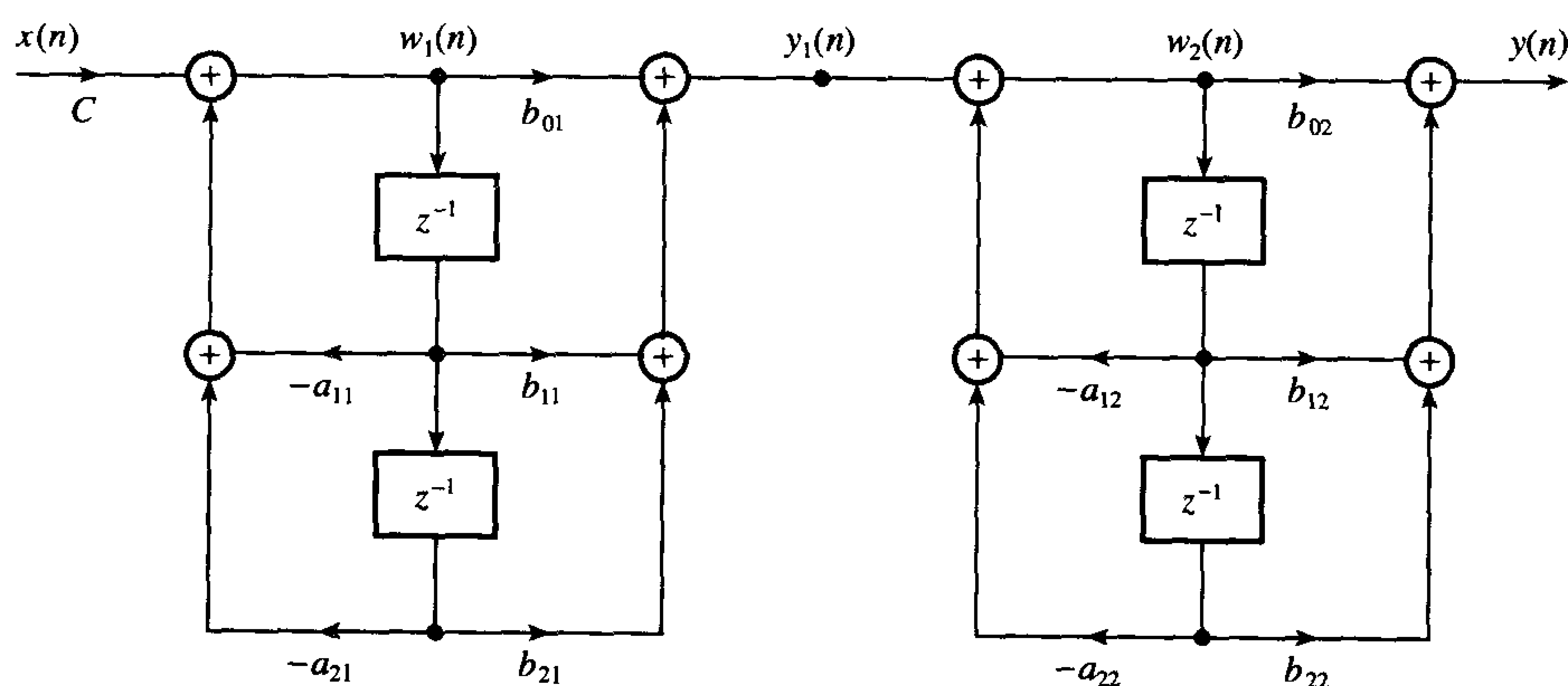
实现结构包含着把一个给定的传递函数$H(z)$转换成一个适当的滤波器结构。框图或流程图经常用来描述滤波器的结构，它们表示了实现数字滤波器的运算过程。采用的结构依赖于滤波器是IIR还是FIR滤波器。

对于IIR滤波器，经常用到的三个结构是直接型、串联型和并联型。直接型是IIR传递函数的直接表示形式。在串联形式里，IIR滤波器的传递函数6.4b式，被因式分解表示成二阶项的乘积形式。在并联形式里，$H(z)$利用部分分式展开，表示成二阶部分的和。为了解释和比较，图6.6给出了一个四阶（也即是$N=4$）的IIR滤波器分别用直接型、串联型和并联型结构表示的框图。在图中同时也给出了描述滤波器结构的相应的一组传递函数和差分方程。

并联和串联结构是IIR里最广泛用到的，因为它们与直接结构相比，可导出更简单的滤波算法，并且对于用有效位实现滤波器而带来的影响不那么敏感。直接结构会遇到严重的系数敏感性问题，特别是对于高阶滤波器，在这些情况下应该避免。



(a) 四阶IIR滤波器的直接实现结构

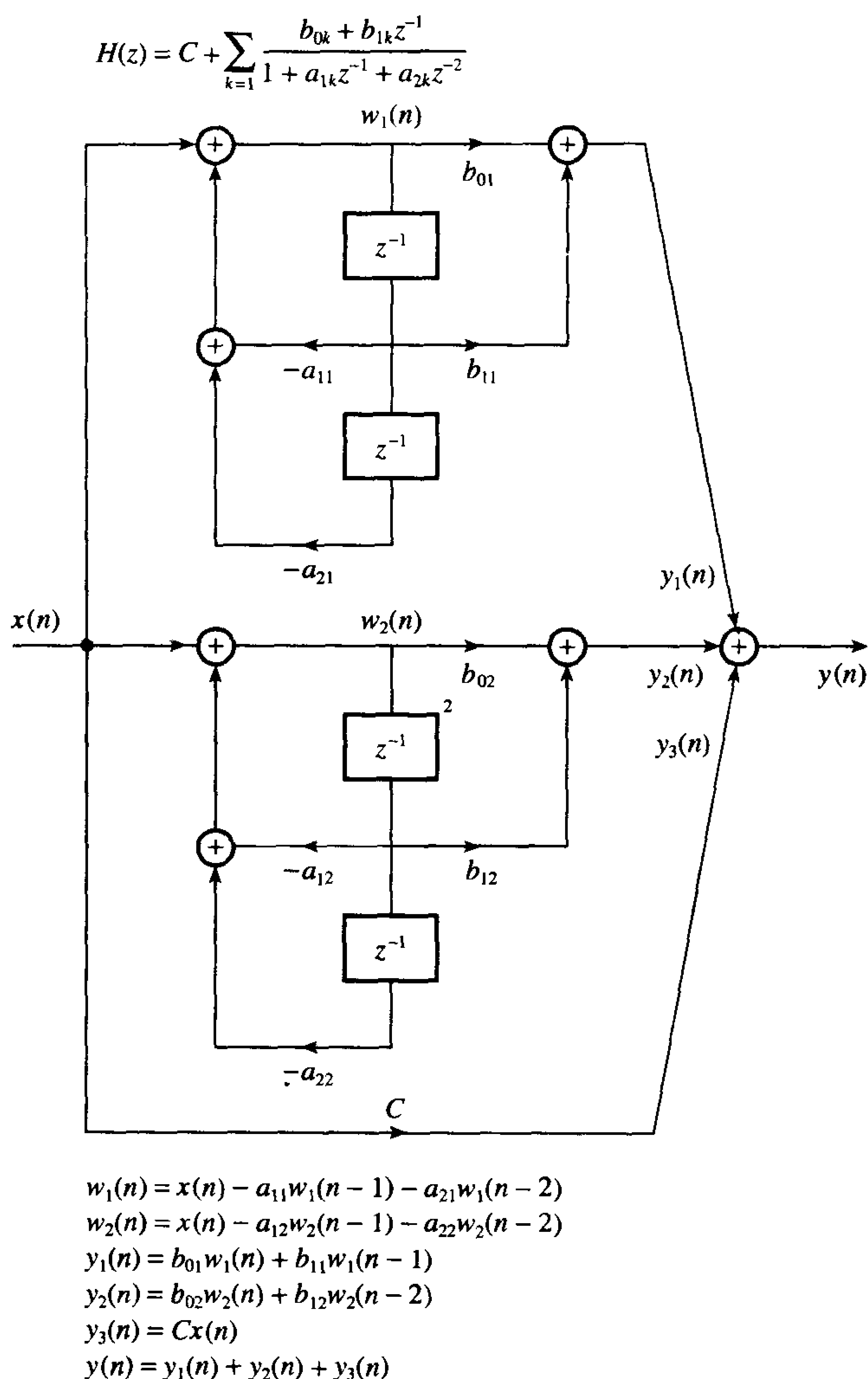


$$H(z)=C\prod_{k=1}^{2}\frac{1+b_{1k}z^{-1}+b_{2k}z^{-2}}{1+a_{1k}z^{-1}+a_{2k}z^{-2}}$$

$$w_1(n)=Cx(n)-a_{11}w_1(n-1)-a_{21}w_1(n-2)$$
$$y_1(n)=b_{01}w_1(n)+b_{11}w_1(n-1)+b_{21}w_1(n-2)$$
$$w_2(n)=y_1(n)-a_{12}w_2(n-1)-a_{22}w_2(n-2)$$
$$y(n)=b_{02}w_2(n)+b_{12}w_2(n-1)+b_{22}w_2(n-2)$$

(b) 四阶IIR滤波器的串联结构

图 6.6　四阶 IIR 滤波器的结构框图



(c) 一个四阶IIR滤波器的并联结构

图6.6（续）　四阶IIR滤波器的结构框图

FIR里最广泛用到的结构是直接型，如图6.7(a)所示，因为它实现起来特别简单。这种形式的FIR有时称为多抽头延迟线（因为它类似于多抽头延迟线）或者横向滤波器。其他两种也用到的FIR结构是频率抽样结构和快速卷积技术，如图6.7(b)和图6.7(c)所示。和横向结构相比，频率抽样结构可以更高效地计算，因为它导出的系数较少，但是它可能实现起来不是那么简单而且要求更大的存储。快速卷积利用了快速傅里叶变换（FFT）的优势，当对信号的功率谱也做了相应的要求时，快速卷积尤其具有吸引力。

其他还有许多数字滤波器的实用性结构，但是大多数仅仅在特定的应用领域内流行。一个例子就是格型结构，它用于语音处理和线性预测应用中。格型结构除了可表示IIR滤波器之外，还可以用来表示FIR滤波器。基本的格型结构可以用一个单一的输入和一对输出来刻画其特性，如图6.8(a)所示。从6.8(a)表示的基本结构推导出的格型滤波器，对于N点FIR滤波器的结构如图6.8(b)所示；对于二阶全极点IIR滤波器（也就是仅仅有分母的系数），它的格型结构如图6.8(c)所示。在例6.5里给出了格型结构更详细的描述。

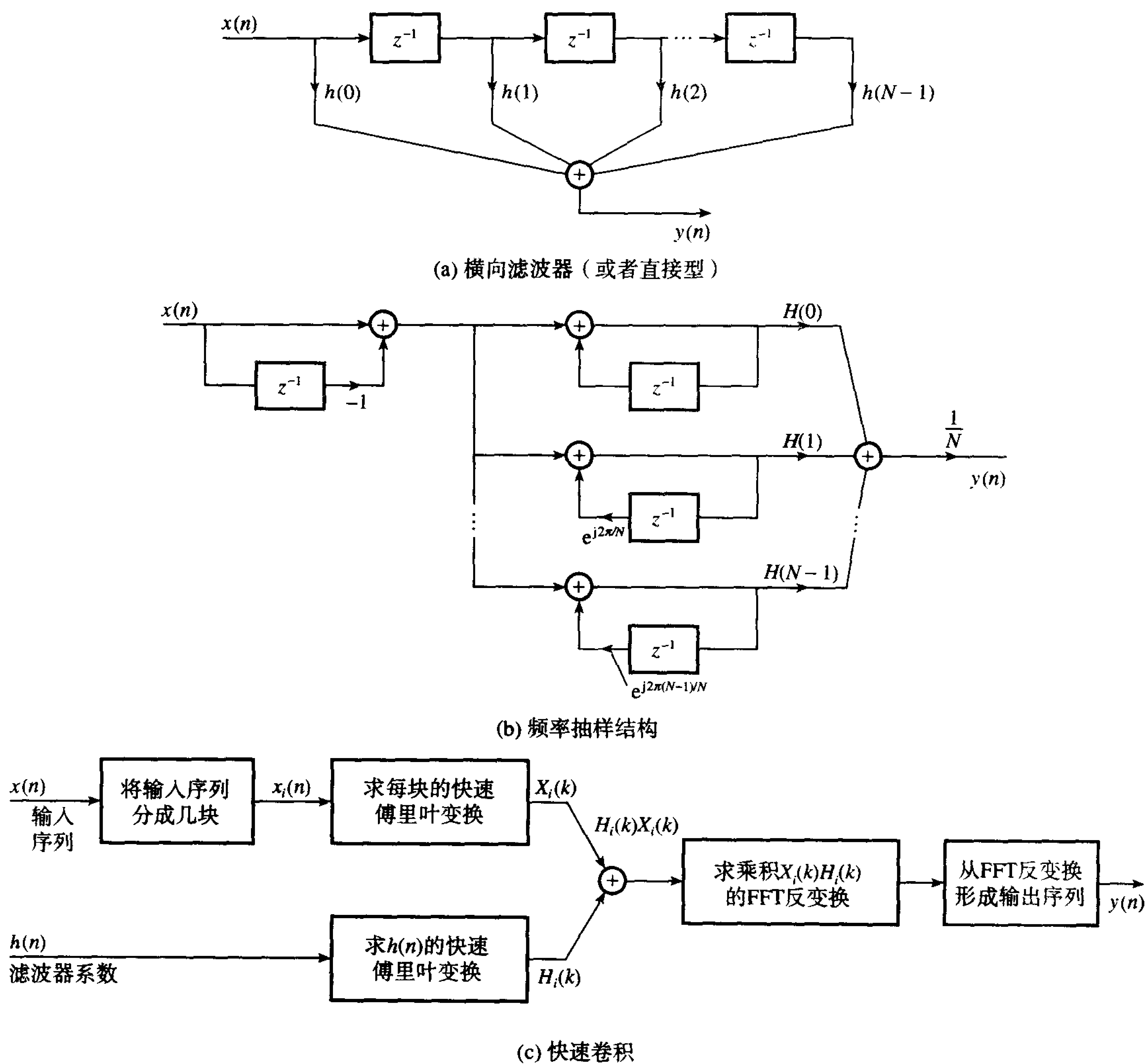


图 6.7　FIR 滤波器的实现结构

总而言之，FIR 和 IIR 滤波器最常用的实现结构如下：

- 横向（直接）　（FIR）
- 频率抽样　（FIR）
- 快速卷积　（FIR）
- 直接型　（IIR）
- 串联　（IIR）
- 并联　（IIR）
- 格型　（IIR 或 FIR）

对于一个给定的滤波器，在以上结构之间应如何选择取决于(i)它是 FIR 还是 IIR，(ii)实现的容易程度，(iii)该结构对有限字长的敏感性如何。FIR 和 IIR 滤波器的实现结构分别在第 7 章和第 8 章里有更充分的讨论。

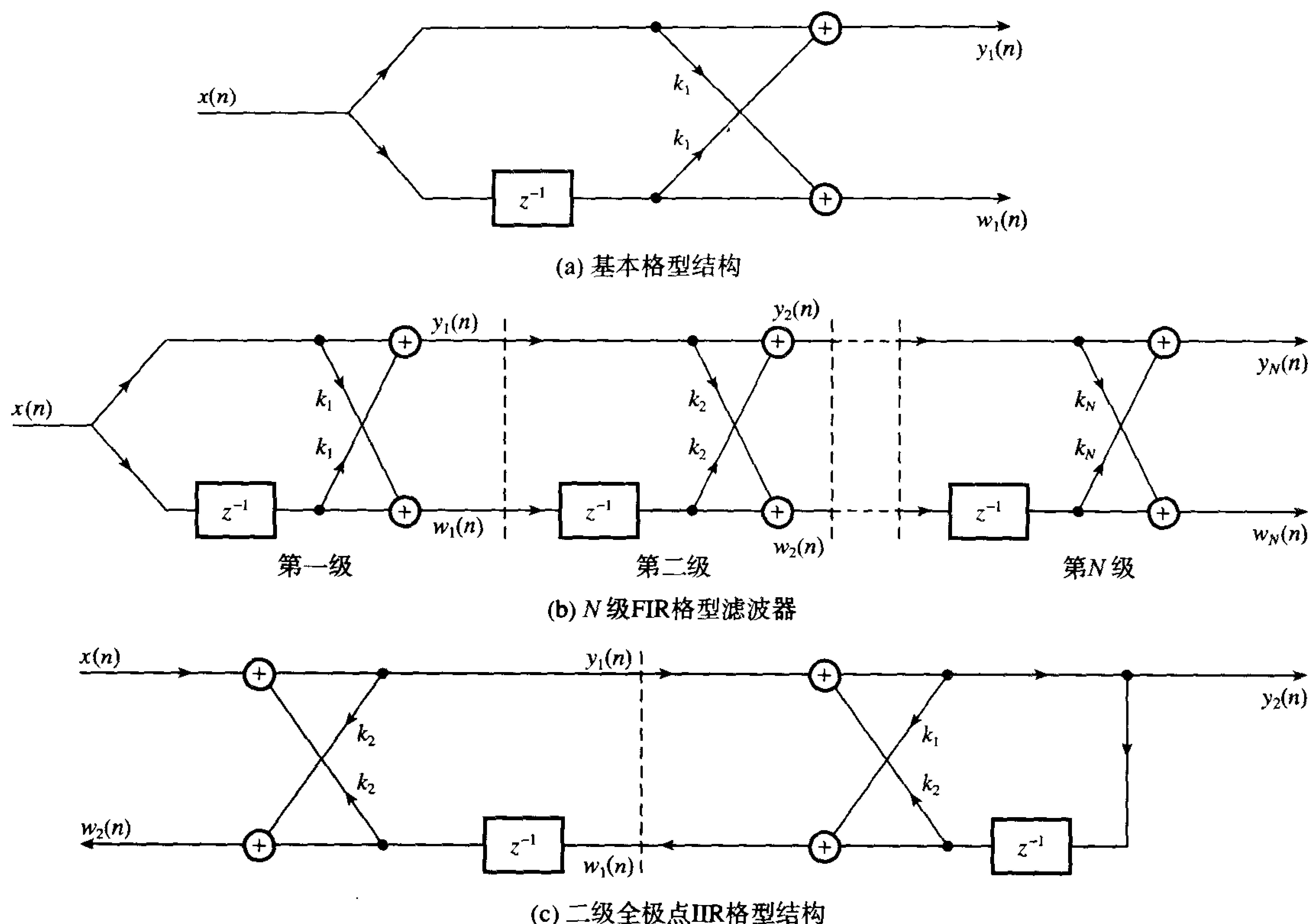


图 6.8　格型结构的各种形式

6.4.4　有限字长效应分析

近似和实现步骤假设为无限或非常高的精度。然而，在实际实现中经常需要用有限位数来表示滤波器系数，典型的是 8 到 16 位。在差分方程中的运算也是有限精度的运算。

有限字长效应会降低滤波器的性能，在某些情况下会导致滤波器不稳定。设计者必须分析这些效应并为滤波器系数选择适当的字长（即位数），而且为滤波器变量（即输入输出抽样值）以及滤波器内的算术运算选择合适的字长。

数字滤波器的性能下降主要有以下主要来源：

- **输入/输出信号的量化**　特别是，由于对输入信号抽样值的量化而带来的 ADC 噪声是最重要的（详情参见第 2 章）。
- **系数量化**　这会在 FIR 和 IIR 滤波器中引起频率响应的偏差，可能引起 IIR 滤波器的不稳定。
- **运算舍入误差**　在应用有限精度运算来执行滤波的过程中会产生需要更多位来表示的结果。当这些结果被量化成允许的字长时，经常是通过舍入，结果引起了舍入噪声。这可能会产生不希望的影响，例如 IIR 滤波器的不稳定。
- **溢出**　当相加的结果超过允许的字长时会发生溢出。它会引起错误的输出样本，以及可能的 IIR 滤波器的不稳定。

滤波器性能下降的程度依赖于(i)用来执行滤波运算的算术类型和字长，(ii)用来量化滤波器系数的方法，以及为变量选择的字长；(iii)滤波器结构。在知道了这些因素之后，设计者就能估计有限字长对滤波器性能的影响，如果有必要可以采用相应的补救措施。

一些效应可能会无关紧要，这依赖于滤波器是如何实现的。例如，在大多数大型计算机上使用高级语言程序时，系数量化和舍入误差可能并不重要。对于实时处理，用有限字长（典型的是8位、12位、16位）来表示输入和输出信号、滤波器系数以及算术运算的结果。在这些情况下，总是需要分析量化对滤波器性能的影响。

在后面的章节详细讨论了量化及其对数字滤波器性能的影响，在第7章讨论了FIR滤波器，在第8章讨论了IIR滤波器。

6.4.5　滤波器的实现

计算了滤波器的系数以后，选择一个合适的结构，并验证在量化系数以及为滤波器变量选择了合适的字长以后性能的下降是可以接受的，接着就必须用软件或者硬件来实现差分方程。不管采用何种实现方法，对于每一个抽样值，滤波器的输出必须根据差分方程（假设是在时域实现）来计算。

考察任意差分方程，我们将发现（6.2式和6.3式）$y(n)$（滤波器输出）的计算仅包括乘法、加法/减法和时延。因此，为了实现一个滤波器，我们需要如下的基本构件：

- 存储滤波器系数的存储器（例如ROM）；
- 存储现在和过去的输入、输出值的存储器（例如RAM），即存储$\{x(n), x(n-1),\ldots\}$和$\{y(n), y(n-1),\ldots\}$；
- 硬件或软件的乘法器；
- 加法器或者算术逻辑单元。

设计者提供这些基本构件，同时也确保了它们的构造对应用是合适的。部件按什么方式构造很大的程度取决于要求的是批处理（即非实时）还是实时处理。在批处理中，整个数据已经在内存中可供使用。例如，在应用中有这样的情况：为了后续分析，实验数据从其他地方获得。在这种情况下，滤波器通常用高级语言实现，在通用计算机上（例如个人电脑）或者大型计算机上运行，其中所有的基本模块都已经构造好了。因此，批处理可描述为一个纯软件实现（尽管设计者可能希望附加硬件以提高处理速度）。

在实时处理情况下，滤波器可能要求：(i)对当前的输入抽样$x(n)$进行运算，在下个输入抽样值到达之前，即在抽样的间隔中，产生当前输出抽样$y(n)$；或者(ii)对一个输入数据进行运算，例如使用FFT算法，在与块长度成比例的周期内产生一个输出数据块。如果抽样率非常高或者滤波器阶数很高，那么实时滤波可能要求快速、专用的硬件。对于大多数音频工作，DSP处理器如DSP56000（摩托罗拉生产）以及TMS320C25（德州仪器生产）是合适的，而且能提供相当大的灵活性。这些处理器在板上配置了所有的基本模块，包括内置硬件乘法器。在某些应用中，标准的8位或者16位的微处理器，例如摩托罗拉6800或68000族能提供相当具有吸引力的、可供选择的实现。除了信号处理硬件之外，设计者还必须对数字硬件提供适当的输入-输出（例如，模拟-数字转换）接口，这依赖于数据源和接收器的类型。对于FIR和IIR滤波器，滤波器实现的详细讨论分别在第7章和第8章中给出。DSP硬件涵盖在第12章、第13章和第14章中。

6.5　说明性的例子

例6.3　讨论数字滤波器设计的五个主要步骤，利用下面的设计问题来说明你的答案。

要求设计一个实时生理学噪声抑制的数字滤波器，滤波器应该满足如下幅度响应性能规范：

通带	0 ~ 10 Hz
阻带	20 ~ 64 Hz
抽样频率	128 Hz
最大通带偏差	< 0.036 dB
阻带衰减	> 30 dB

其他重要的要求：

(1) 带内信号的各分量之间的间谐波关系失真最小是高度期望的；

(2) 滤波的可用时间是有限的，滤波是一个大的过程的一部分；

(3) 滤波器要用德州仪器生产的 TMS32010 DSP 处理器实现，模拟输入被量化到 12 位。

解：

这个滤波器被设计用在某个生物医学信号的处理项目中。我们这里仅给出设计的概要性的讨论。详细的讨论将推迟到第 7 章，在那里涵盖了 FIR 滤波器完整的设计方法。

(1) **要求的性能规范**　如前面所讨论的那样，设计者必须给出滤波器的恰当任务和性能要求以及其他任何重要的约束。在这个例子中已经给出了这些内容。

(2) **计算合适的系数**　通过最佳方法法得到一个线性相位 FIR 滤波器的系数，能最好地达到最小失真和有限处理时间的要求。

(3) **滤波器结构选择**　横向结构可推导出最有效的用 TMS32010 处理器实现的结构。

(4) **滤波器有限字长效应分析**　因为要用到TMS32010处理器，采用定点运算，为了效率起见，每个系数用 16 位表示。FIR 滤波器性能的下降可能来自于输入信号的量化、系数的量化、舍入和溢出误差。应该做一个检验来确保字长足够长。对于本题的这种情况，有限字长效应的分析表明了输入量化噪声以及因为系数量化而引起的频率响应的偏差两者都是微不足道的。利用 TMS32010 的 32 位加法器来对系数数据的乘积求和，仅对最终的和值做舍入，这样会把舍入误差降低到可忽略的程度。为了避免溢出，每一个系数在量化成16位之前应该除以 $\sum_{k=0}^{N-1}|h(k)|$。

(5) **实现**　用必要的输入 / 输出接口来设计和构造基于 TMS32010 的硬件（如果它还不存在）。接着编写 TMS32010 程序来处理 I/O 协议，以及对每一个新的输入 $x(n)$ 计算滤波器输出 $y(n) = \sum_{k=0}^{N-1} h(k)x(n-k)$。

例 6.4　一个模拟滤波器要被转换成一个等价的数字滤波器，抽样频率是 256 Hz。模拟滤波器的传递函数是

$$H(s) = \frac{1}{s^3 + 2s^2 + 2s + 1}$$

(1) 求数字滤波器的合适的系数。

(2) 假设利用串联结构来实现数字滤波器，画出合适的实现方框图，并建立差分方程。

(3) 对于并联结构重复(2)。

解：

(1) 为了保留模拟函数的幅度响应，利用双线性转换法来求滤波器系数。对于模拟传递函数应用双线性转换法产生如下的传递函数：

$$H(z)=\frac{0.1432(1+3z^{-1}+3z^{-2}+z^{-3})}{1-0.1801z^{-1}+0.3419z^{-2}-0.0165z^{-3}}$$

(2) 对于串联实现结构，利用部分因式分解得到 $H(z)$为

$$H(z)=0.1432\frac{1+2z^{-1}+z^{-2}}{1-0.1307z^{-1}+0.3355z^{-2}}\frac{1+z^{-1}}{1-0.0490z^{-1}}$$

框图表示如图 6.9 所示，对应的差分方程组为

$$w_1(n)=0.1432x(n)+0.1307w_1(n-1)-0.3355w_1(n-2)$$
$$y_1(n)=w_1(n)+2w_1(n-1)+w_1(n-2)$$
$$w_2(n)=y_1(n)+0.049w_2(n-1)$$
$$y_2(n)=w_2(n)+w_2(n-1)$$

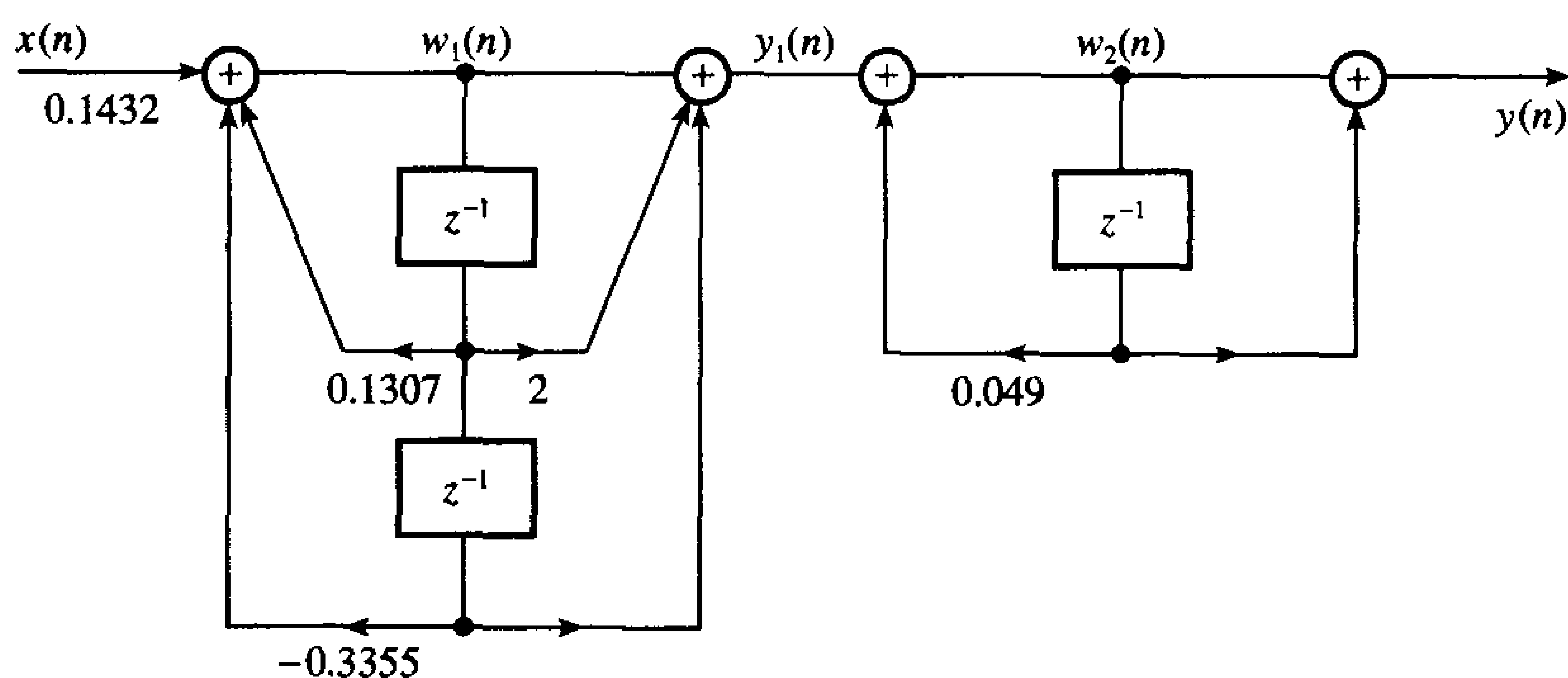


图 6.9　串联实现结构的框图

(3) 对于并联结构，$H(z)$用部分分式（详情见第 4 章和第 8 章）表示为

$$H(z)=\frac{1.2916-0.084\,07z^{-1}}{1-0.131z^{-1}+0.3355z^{-2}}+\frac{7.5268}{1-0.049z^{-1}}-8.6753$$

并联结构的框图如图 6.10 所示，对应的差分方程组为

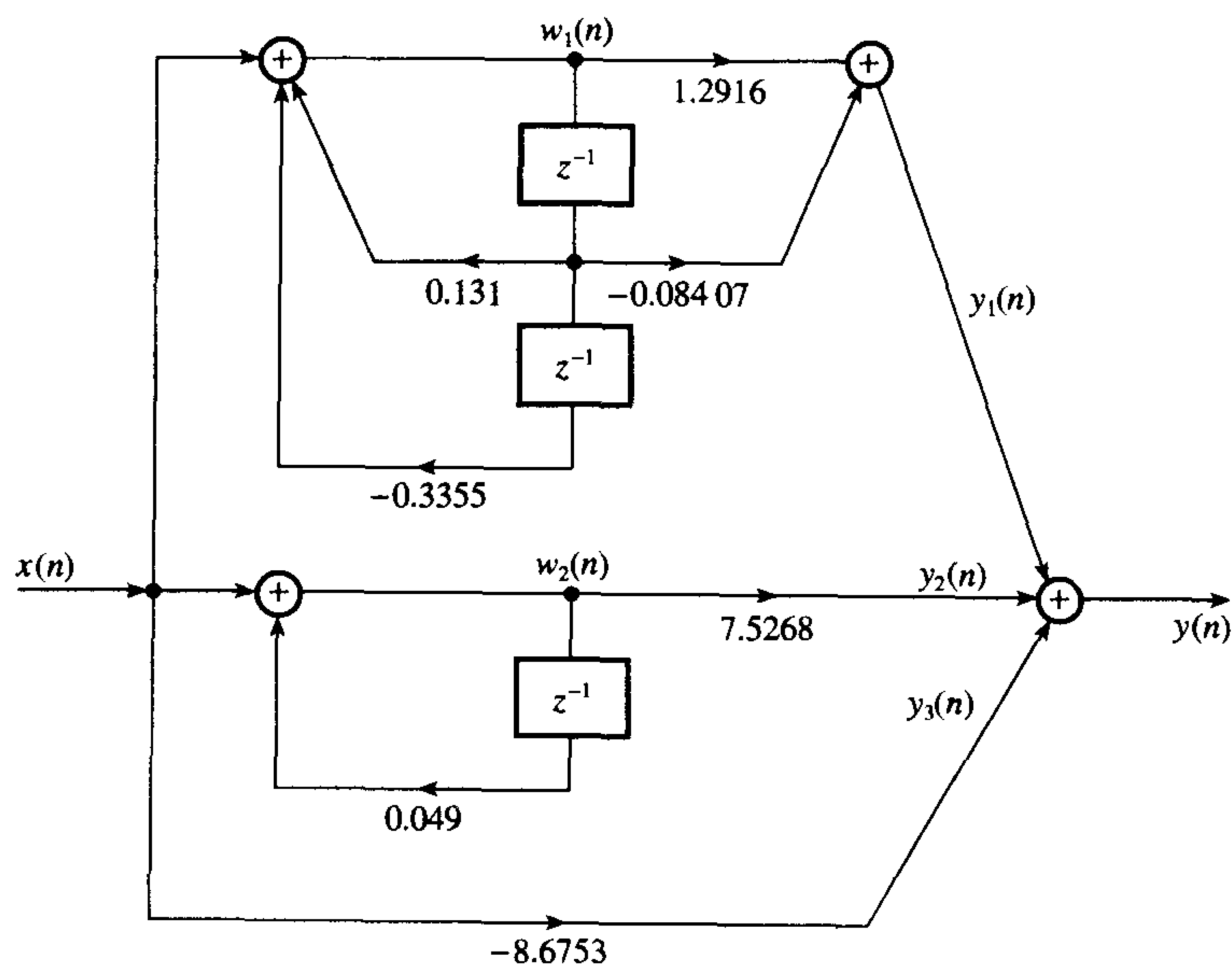


图 6.10　并联结构的框图

$$w_1(n) = x(n) + 0.131w_1(n-1) - 0.3355w_1(n-2)$$
$$y_1(n) = 1.2916w_1(n) - 0.084\,07w_1(n-1)$$
$$w_2(n) = x(n) + 0.049w_2(n-1)$$
$$y_2(n) = 7.5268w_2(n)$$
$$y_3(n) = -8.6753x(n)$$
$$y(n) = y_1(n) + y_2(n) + y_3(n)$$

例 6.5 FIR 滤波器的传递函数为

$$H(z) = 1 - 1.3435z^{-1} + 0.9025z^{-2}$$

对下面的每一种情况画出其实现框图：

(1) 横向结构；
(2) 二级格型结构。

计算格型结构的系数值。

解：

(1) 根据传递函数，横向结构的框图如图 6.11 所示。横向结构的输入和输出为

$$y(n) = x(n) + h(1)x(n-1) + h(2)x(n-2) \tag{6.6}$$

(2) 滤波器的一个二级格型结构如图 6.12 所示。该结构的输入输出关系为

$$\begin{aligned} y_2(n) &= y_1(n) + k_2w_1(n-1) \\ &= x(n) + k_1(1+k_2)x(n-1) + k_2x(n-2) \end{aligned} \tag{6.7a}$$

$$w_2(n) = k_2x(n) + k_1(1+k_2)x(n-1) + x(n-2) \tag{6.7b}$$

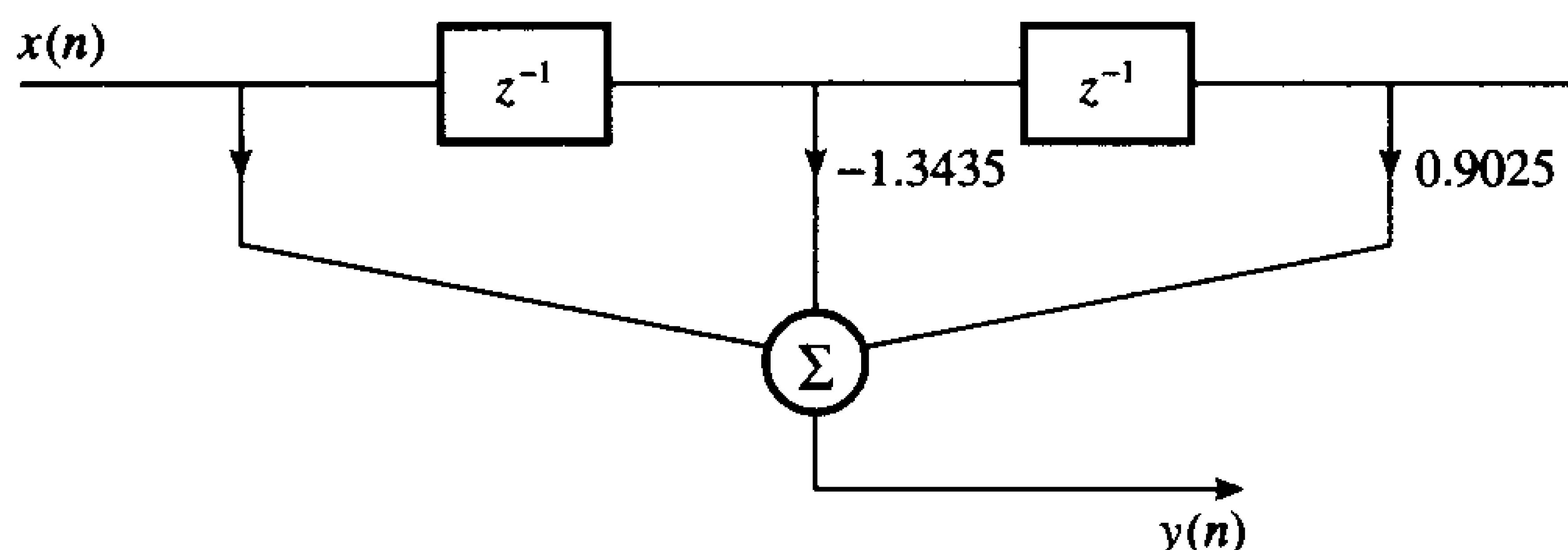


图 6.11　横向结构的框图

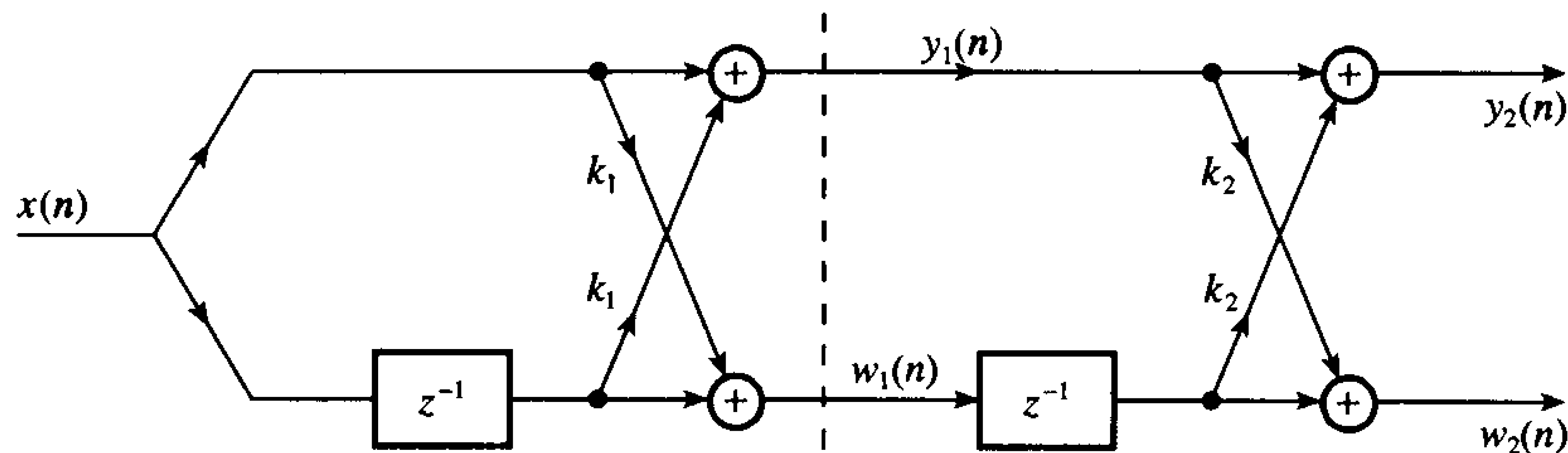


图 6.12　二级格型结构

比较 6.6 式和 6.7a 式，采用待定系数法，我们有

$$k_1 = \frac{h(1)}{1+h(2)} \qquad k_2 = h(2)$$

从而得到 $k_2 = 0.9025$，$k_1 = -1.3435/(1+0.9025) = -0.7062$。

注意，$y_2(n)$和$w_2(n)$的系数（6.7a式和6.7b式）除了一个是用反序写的之外都是相同的。这是FIR格型结构的特征。有关格型结构的更详细的说明，包括用递归技术把FIR或者IIR滤波器的系数转换成等价的格型结构的系数，在一些著作中都已经给出（例如，参见Proakis and Manolakis, 1992）。

6.6 小结

数学算法的软件或硬件实现称为数字滤波器，该算法接受数字信号作为输入，产生另一个数字信号作为输出，信号的波形和/或幅度及相位的响应按特定的方式进行了修改。在许多应用中，数字滤波器的应用优先于模拟滤波器，因为它们能更好地满足幅度和相位规范，并且能消除通常模拟滤波器具有的温度和电压的漂移。

在这一章我们给出了FIR和IIR滤波器设计的一般框架，包括从性能规范到实现。设计这些滤波器的简单的逐步指南过程包括五个主要步骤: (i)滤波器性能规范; (ii)合适的滤波器系数的计算; (iii)利用合适的结构的滤波器实现；(iv)量化滤波器的系数，对变量选择合适的字长，分析可能导致的误差; (v)实现，即涉及到硬件或者处理器（处理器对输入数据进行实际的滤波）中的软件编程。

习题

6.1　假设6.4.2节中给出的6种计算滤波器系数的方法都是可用的。在下面的应用中，阐述并论证应该使用哪种方法：

(1) 数字通信系统中的相位（延时）均衡；

(2) 仿真一个模拟系统；

(3) 一个高吞吐率的噪声抑制系统，要求锐利的幅度频率响应滤波器；

(4) 图像处理；

(5) 高质量的数字音频处理；

(6) 具有最小失真的实时生物医学信号处理。

6.2　下面的传递函数表示两个不同的滤波器，它们满足相同的幅度频率响应规范：

(1) $H(z) = \dfrac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \times \dfrac{b_3 + b_4 z^{-1} + b_5 z^{-2}}{1 + a_3 z^{-1} + a_4 z^{-2}}$

其中

$b_0 = 3.136\,362 \times 10^{-1}$

$b_1 = -5.456\,657 \times 10^{-2}$

$b_2 = 4.635\,728 \times 10^{-1}$

$b_3 = -5.456\,657 \times 10^{-2}$

$b_4 = 3.136\,362 \times 10^{-1}$

$b_5 = 4.635\,728 \times 10^{-1}$

$a_1 = -8.118\,702 \times 10^{-1}$

$a_2 = 3.339\,288 \times 10^{-1}$

$a_3 = -2.794\,577 \times 10^{-1}$

$a_4 = 3.030\,631 \times 10^{-1}$

(2) $H(z)=\sum_{k=0}^{22} h_k z^{-k}$

其中

$h_0 = \ \ 0.398\,264\,80 \times 10^{-1} = h_{22}$

$h_1 = -0.168\,743\,80 \times 10^{-1} = h_{21}$

$h_2 = \ \ 0.347\,811\,30 \times 10^{-1} = h_{20}$

$h_3 = \ \ 0.120\,528\,90 \times 10^{-1} = h_{19}$

$h_4 = -0.447\,318\,60 \times 10^{-1} = h_{18}$

$h_5 = \ \ 0.278\,946\,10 \times 10^{-1} = h_{17}$

$h_6 = -0.875\,733\,60 \times 10^{-1} = h_{16}$

$h_7 = -0.909\,720\,60 \times 10^{-1} = h_{15}$

$h_8 = -0.156\,675\,50 \times 10^{-1} = h_{14}$

$h_9 = -0.284\,995\,60 \times 10^{0} = h_{13}$

$h_{10} = \ \ 0.740\,350\,30 \times 10^{-1} = h_{12}$

$h_{11} = \ \ 0.623\,495\,60 \times 10^{0}$

对于每一个滤波器,

(a) 说明它是 FIR 还是 IIR 滤波器；

(b) 用框图形式来表示滤波运算，并写出差分方程；

(c) 确定并解释计算量和存储量。

6.3 要求一个数字滤波器,它能从存储在主机内存里的胎儿的心电图数据中移去其主要分量。数据被数字化到 12 位的精度。

滤波器的性能规范包括：

在主频处的衰减	> 50 dB
通带波纹	< 0.05 dB
通带边沿	0 ~ 0.09 以及 0.11 ~ 0.5（经过归一化）
抽样频率	500 Hz

这个滤波器应该用高级语言实现，它可以从主分析程序调用。滤波器引起的任何信号失真都应该保持到最小程度，因为重要的 ECG（心电图）波形很容易被破坏。

充分地讨论设计中包含的问题,指出设计者可能面临的各种选择,并给出建议和理由。

6.4 要求一个数字滤波器用来对原始的胎儿心电图（ECG）数据进行预处理。ECG 是心脏的电活动，预处理的目的是为了在基线偏移、电源干扰、子宫收缩、胎儿和母亲运动等情况下能较容易地检测出胎儿的心跳。基线摇摆和移动的伪像占用的频率范围为0 ~ 10 Hz，而电源干扰的频率集中在 50 Hz 或 60 Hz（与国家有关）。需要用来检测心跳的 ECG 里的大部分能量被认为位于 5 Hz 和 50 Hz 之间。

假设你已经有了胎儿的ECG数据，这些数据是在0.05 ~ 100 Hz间模拟滤波，并且以 500 抽样值 / 秒以及 8 位的分辨率进行数字化。

(1) 假设采用 IIR 滤波器，确定一组数字滤波器规范，论证你的结论。

(2) 对 FIR 滤波器重复(1)。

对于上述情况中的应用，这两个滤波器的哪一个最好？为什么？

6.5 在一个选择了现有的有源滤波器的特定的语音传输系统中，利用一个数字滤波器提供抗混叠滤波。现在对这个系统的模拟输入信号在利用一个具有如下性能规范的有源滤波器限带之后以 8 kHz 的频率抽样：

通带	0 ~ 3.4 kHz
阻带边沿频率	8 kHz
阻带内的衰减	30 dB
在 4 kHz 处的衰减	14 dB
通带波纹	< 0.1 dB

借助框图进行讨论，通过数字滤波，性能规范是怎样得到满足的。指定我们将用到的滤波器的类型以及它的特性，给出理由。

6.6 对于一个通信系统需要从远程发射机接收到的含噪声的数据中，利用数字滤波技术恢复出时钟频率，使得数据可以可靠地提取出来。远程发射机的时钟频率已知是2.048 MHz。讨论本任务的合适的数字滤波器特性，并指定它的传递函数。

6.7 一个格型 FIR 滤波器的系数是 $k_1 = -0.266$，$k_2 = 0.69$，画出格型滤波器的实现结构图，计算滤波器的冲激响应的系数，并画出等效的横向结构的框图。

6.8 一个二阶 IIR 数字滤波器用如下的传递函数来描述其特性：

$$H(z) = \frac{1}{1 - 0.9z^{-1} + 0.81z^{-2}}$$

对如下结构分别画出实现结构图：

(1) 直接型；

(2) 格型。

从给定的传递函数求出格型结构的系数。

参考文献

Proakis J.G. and Manolakis D.G. (1992) *Digital Signal Processing*, 2nd edn. New York: Macmillan.

参考书目

DeFatta D.J., Lucas J.G. and Hodgkiss W.S. (1988) *Digital Signal Processing*. New York: Wiley.

Elliott D.F. (ed.) (1987) *Handbook of Digital Signal Processing*. London: Academic Press.

Oppenheim A.V. and Schafer R.W. (1975) *Digital Signal Processing*. Englewood Cliffs NJ: Prentice-Hall.

Parks T.W. and Burrus C.S. (1987) *Digital Filter Design*. New York: Wiley.

Rabiner L.R. and Gold B. (1975) *Theory and Application of Digital Signal Processing*. Englewood Cliffs NJ: Prentice-Hall.

Rabiner L.R., Cooley J.W., Helms H.D., Jackson L.B., Kaiser J.F., Rader C.M., Schafer R.W., Steiglitz K. and Weinstein C.J. (1972) Terminology in digital signal processing. *IEEE Trans. Audio Electroacoustics*, **20** (December), 322–37.

Taylor F.J. (1983) *Digital Filter Design Handbook*. New York: Dekker.

第7章　有限冲激响应（FIR）滤波器设计

本章从技术规范到有限字长效应的分析（通过系数计算）和实现来介绍FIR滤波器的设计，利用处理过的例子来说明各设计阶段并巩固重要概念。一个完整意义上的滤波器设计应包括对所有阶段是如何恰当地衔接起来的说明，以及能给设计滤波器的人们提供指导。基于PC的MATLAB程序可以在网上找到（详情请参见前言），另外指导手册的CD上含有MATLAB和C语言程序供读者使用，并且可以用来重现书中的结果，或者为读者设计专用滤波器服务。

7.1　引言

首先，在讨论FIR滤波器设计之前简要叙述一下FIR滤波器的重要特征。

7.1.1　FIR滤波器关键特征的概述

(1) 下面两个公式刻画出基本的FIR滤波器

$$y(n)=\sum_{k=0}^{N-1}h(k)x(n-k) \tag{7.1a}$$

$$H(z)=\sum_{k=0}^{N-1}h(k)z^{-k} \tag{7.1b}$$

式中$h(k)$（$k=0, 1,..., N-1$），是滤波器的冲激响应系数。$H(z)$是滤波器的传递函数，N是滤波器长度，即滤波器系数的数目。7.1a式是FIR时域差分方程，它使用非递归形式描述了FIR滤波器：当前输出信号$y(n)$只是过去和当前的输入值$x(n)$的函数。当FIR滤波器利用这种形式实现时，即直接根据7.1a式进行计算，那么滤波器总是稳定的。7.1b式是滤波器的传递函数，它提供了分析滤波器的一种方法，例如评估频率响应。

(2) FIR滤波器有精确的线性相位响应，其含义将在下一节讨论。

(3) FIR滤波器实现起来非常简单，目前所有可用的DSP处理器具有适合于FIR滤波的结构。非递归FIR滤波器受有限字长的影响要比IIR滤波器小。递归FIR滤波器有着重要的计算优势（详情请参见7.7节）。

无论何时使用上述优点的哪一个，尤其是线性相位优点，都应使用FIR滤波器。在FIR滤波器和IIR滤波器之间进行选择时所要考虑的问题在6.3节给出。

7.1.2　线性相位响应及其含义

精确的线性相位响应能力是FIR滤波器最重要的特性之一。因此，我们将着重考察这一特性。当信号通过滤波器时，其振幅和相位被修正，信号修正的程度和性质依赖于滤波器的振幅和相位特性。滤波器的相位延迟和群延迟（或称包络线延迟）为滤波器如何修正信号相位提供了一种有用的度量。如果我们考虑由几个频率部分组成的信号（如语音波形和调制信号），那么滤波器的相位延迟是信号的各个频率分量通过滤波器所经历的时延量。另一方面，群延迟是合成信号在每个频率经历的平均时延。从数学上讲，相位延迟等于相位角除以频率的负数，而群延迟是相位相对于频率的导数的负数：

$$T_p = -\theta(\omega)/\omega \tag{7.2a}$$

$$T_g = -\mathrm{d}\theta(\omega)/\mathrm{d}\omega \tag{7.2b}$$

具有非线性相位特性的滤波器，当信号通过滤波器时会引起相位失真。这是因为信号中的每个频率分量的延迟大小于与频率不成比例，因此改变了它们之间的谐波关系。这样的失真在许多应用中是不希望出现的，如乐曲、数字传输、录像和生物医学。对感兴趣频段使用具有线性相位响应的滤波器可以消除相位失真。

如果滤波器的相位响应满足下面两个关系的一个，该滤波器就称为具有线性相位响应的滤波器：

$$\theta(\omega) = -\alpha\omega \tag{7.3a}$$

$$\theta(\omega) = \beta - \alpha\omega \tag{7.3b}$$

其中α和β是常量。若滤波器满足7.3a式给定的条件，则它具有恒定的群延迟和恒定的相位延迟响应。可以证明，为了满足条件7.3a，滤波器的冲激响应必须具有正的对称性，这种情况下的相位响应简单地就是滤波器长度的函数：

$$h(n) = h(N-n-1), \quad \begin{cases} n = 0, 1, \ldots, (N-1)/2 & (N\text{为奇数}) \\ n = 0, 1, \ldots, (N/2)-1 & (N\text{为偶数}) \end{cases}$$

$$\alpha = (N-1)/2$$

当只满足7.3b式给出的条件时，滤波器将只有恒定的群延迟。这种情况下，滤波器的冲激响应是负对称的：

$$h(n) = -h(N-n-1)$$

$$\alpha = (N-1)/2$$

$$\beta = \pi/2$$

线性相位FIR滤波器是FIR滤波器的一个重要类型，它们有一组独特的特性，这些特性对我们如何设计和实现滤波器有一定影响。我们将举例探讨其中的一些特性。

例7.1

(1) 简要地论述实现具有线性相位特性的数字滤波器所需要的条件，以及具有这一特性的滤波器的优点。

(2) FIR数字滤波器的冲激响应$h(n)$定义在区间$0 \leqslant n \leqslant N-1$上。证明：如果$N=7$，且$h(n)$满足对称条件：

$$h(n) = h(N-n-1)$$

则称该滤波器具有线性相位特性。

(3) 若$N=8$，重复(2)。

解：

(1) 滤波器具有线性相位特性的充要条件是它的冲激响应必须是对称的（Rabiner and Gold, 1975）：

$$h(n) = h(N-1-n) \text{ 或者 } h(n) = -h(N-1-n)$$

对于非递归FIR滤波器，系数的存储空间和算术运算次数几乎减少了2倍。对于递归滤波器，系数转变为简单整数，提高了处理速度。在线性相位滤波器中，所有频率分量通过滤波器时会遇到同样大小的延迟，即没有相位失真。

(2) 利用对称条件，当 $N=7$ 时，我们得出：

$$h(0)=h(6);\ h(1)=h(5);\ h(2)=h(4)$$

该滤波器的频率响应 $H(\omega)$ 由 7.1b 式通过令 $z=e^{j\omega T}$ 得到：

$$\begin{aligned}H(\omega)&=H(e^{j\omega T})\\&=\sum_{k=0}^{6}h(k)e^{-jk\omega T}\\&=h(0)+h(1)e^{-j\omega T}+h(2)e^{-j2\omega T}+h(3)e^{-j3\omega T}+h(4)e^{-j4\omega T}\\&\quad+h(5)e^{-j5\omega T}+h(6)e^{-j6\omega T}\\&=e^{-j3\omega T}[h(0)e^{j3\omega T}+h(1)e^{j2\omega T}+h(2)e^{j\omega T}+h(3)+h(4)e^{-j\omega T}\\&\quad+h(5)e^{-j2\omega T}+h(6)e^{-j3\omega T}]\end{aligned}$$

利用对称条件，我们将系数相同的项合并：

$$\begin{aligned}H(\omega)&=e^{-j3\omega T}[h(0)(e^{j3\omega T}+e^{-j3\omega T})+h(1)(e^{j2\omega T}+e^{-j2\omega T})\\&\quad+h(2)(e^{j\omega T}+e^{-j\omega T})+h(3)]\\&=e^{-j3\omega T}[2h(0)\cos(3\omega T)+2h(1)\cos(2\omega T)\\&\quad+2h(2)\cos(\omega T)+h(3)]\end{aligned}$$

如果我们令 $a(0)=h(3)$，$a(k)=2h(3-k)$（$k=1,2,3$），那么 $H(\omega)$ 可以用简洁的形式表示为

$$H(\omega)=\sum_{k=0}^{3}a(k)\cos(n\omega T)e^{-j3\omega T}=\pm|H(\omega)|e^{j\theta(\omega)}$$

其中

$$\pm|H(\omega)|=\sum_{k=0}^{3}a(k)\cos(n\omega T);\ \theta(\omega)=-3\omega T$$

显然，相位响应是线性的。

(3) 在这种情况下，由对称条件可推出：

$$h(0)=h(7);\ h(1)=h(6);\ h(2)=h(5);\ h(3)=h(4)$$

类似地，利用对称条件，我们有

$$\begin{aligned}H(\omega)&=e^{-j7\omega T/2}[h(0)(e^{j7\omega T/2}+e^{-j7\omega T/2})+h(1)(e^{j5\omega T/2}+e^{-j5\omega T/2})\\&\quad+h(2)(e^{j3\omega T/2}+e^{-j3\omega T/2})+h(3)(e^{j\omega T/2}+e^{-j\omega T/2})]\\&=e^{-j7\omega T/2}[2h(0)\cos(7\omega T/2)+2h(1)\cos(5\omega T/2)\\&\quad+2h(2)\cos(3\omega T/2)+2h(3)\cos(\omega T/2)]\\&=\pm|H(\omega)|e^{j\theta(\omega)}\end{aligned}$$

其中

$$\pm|H(\omega)|=\sum_{k=1}^{4}b(k)\cos[\omega(k-1/2)];\ \theta(\omega)=-(7/2)\omega T$$

$$b(k)=2h(N/2-k),\quad k=1,2,\ldots,N/2$$

FIR 滤波器的上述结果概括在表 7.1 中。

表 7.1　四种线性相位 FIR 滤波器的关键点概要

冲激响应对称性	系数点数 N	频率响应 $H(\omega)$	线性相位类型
正对称 $h(n)=h(N-1-n)$	奇数	$e^{-j\omega(N-1)/2}\sum_{n=0}^{(N-1)/2}a(n)\cos(\omega n)$	1
	偶数	$e^{-j\omega(N-1)/2}\sum_{n=1}^{N/2}b(n)\cos[\omega(n-\frac{1}{2})]$	2
负对称 $h(n)=-h(N-1-n)$	奇数	$e^{-j[\omega(N-1)/2-\pi/2]}\sum_{n=1}^{(N-1)/2}a(n)\sin(\omega n)$	3
	偶数	$e^{-j[\omega(N-1)/2-\pi/2]}\sum_{n=1}^{N/2}b(n)\sin[\omega(n-\frac{1}{2})]$	4

$a(0)=h[(N-1)/2]$; $a(n)=2h[(N-1)/2-n]$
$b(n)=2h(N/2-n)$

7.1.3　线性相位滤波器类型

现在有四种线性相位滤波器，取决于N是偶数还是奇数、$h(n)$是正对称还是负对称。在上面的例子中考虑到了线性相位滤波器四种类型中的两种。图 7.1 说明了线性相位 FIR 滤波器的四种类型在冲激响应上的区别。表 7.1 总结了它们的关键特征。

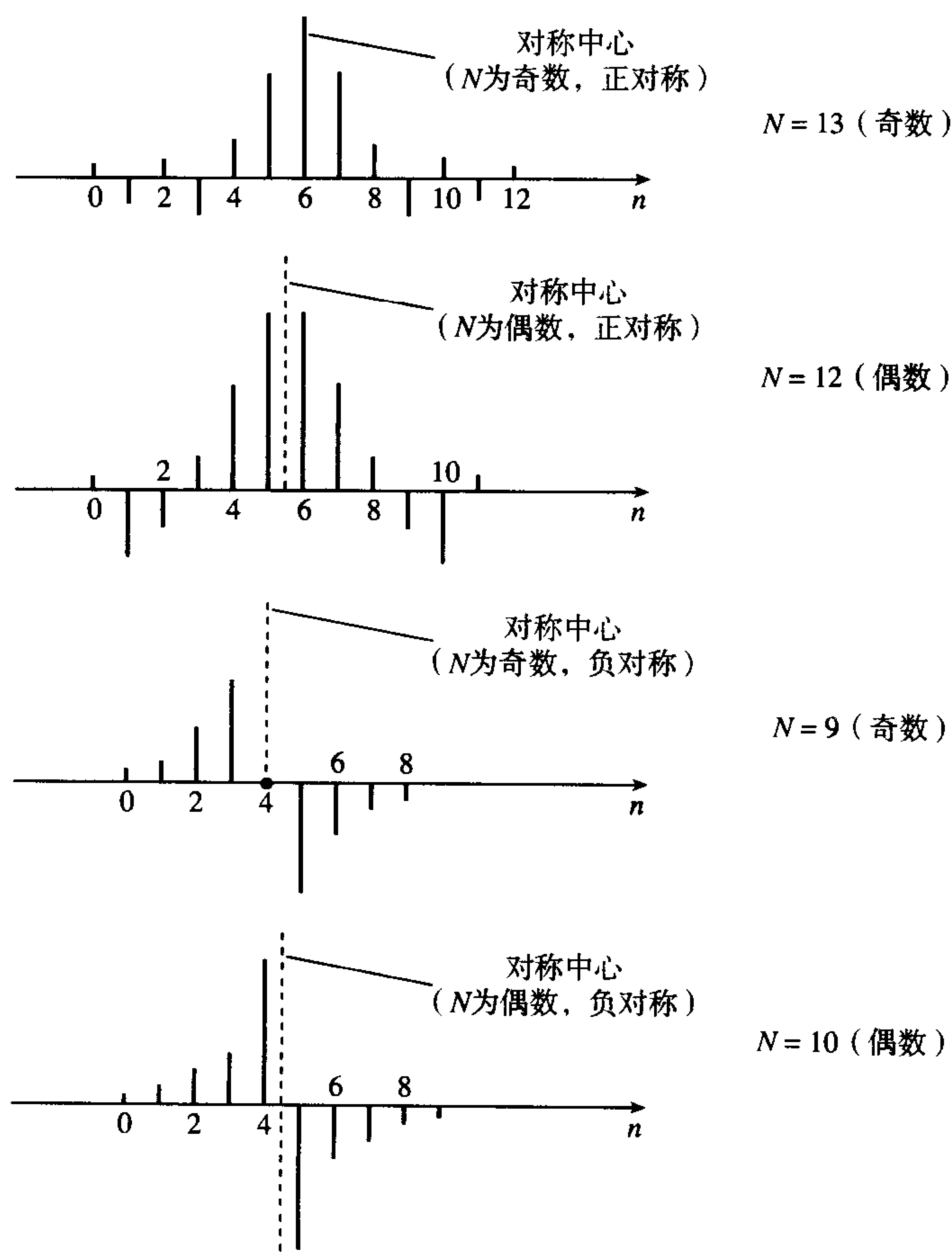


图 7.1　四种类型的线性相位滤波器的冲激响应比较

类型2滤波器（正对称，偶数长度）的频率响应在$f=0.5$处（半抽样频率，且所有频率都用抽样频率归一化）总是为零；参见习题7.1。因此，这种类型的滤波器不适合作为高通滤波器。类型3和类型4（两种都是负对称）每种都引入了90°的相移。频率响应在$f=0$处总是为零，它们不适合作为低通滤波器。类型1是四种类型中最通用的，类型3和类型4经常用来设计微分器和希尔伯特（Hilbert）变换器，因为它们每个都能提供90°的相移。

相位延迟（对类型1和类型2的滤波器）或群延迟（对所有四种类型的滤波器）可以根据滤波器的系数的数目来表达，所以也能够进行修正，以便给出零相位或群延迟响应，注意到这一点是很重要的。例如，对于类型1和类型2的滤波器，给出的相位延迟为

$$T_p=\left(\frac{N-1}{2}\right)T \tag{7.4a}$$

而对类型3和类型4的滤波器，群延迟为

$$T_p=\left(\frac{N-1-\pi}{2}\right)T \tag{7.4b}$$

其中T为抽样周期。

7.2 FIR滤波器设计

如第6章讨论的那样，数字滤波器设计包括5步，即：

(1) **滤波器技术规范** 包括滤波器类型（例如低通滤波器）的确定，期望的幅度和相位响应和我们可以接受的公差，以及抽样频率和输入数据的字长。

(2) **系数计算** 在这一步，我们确定满足第一步给出的技术规范的传递函数$H(z)$的系数。影响我们系数计算方法选择的因素有多个，(1)中的关键要求是最重要的。

(3) **实现（realization）** 这包括将(2)中的传递函数转换为合适的滤波器网络或结构。

(4) **有限字长效应分析** 这里我们分析滤波器系数和输入数据量化的影响，以及用固定字长执行滤波的运算对滤波器性能的影响。

(5) **工程实现（implementation）** 这包括编写软件代码和生产硬件并且执行实际的滤波。

图7.2是对这相关联的5步的总结。现在以图为例实现FIR滤波器设计的5步。

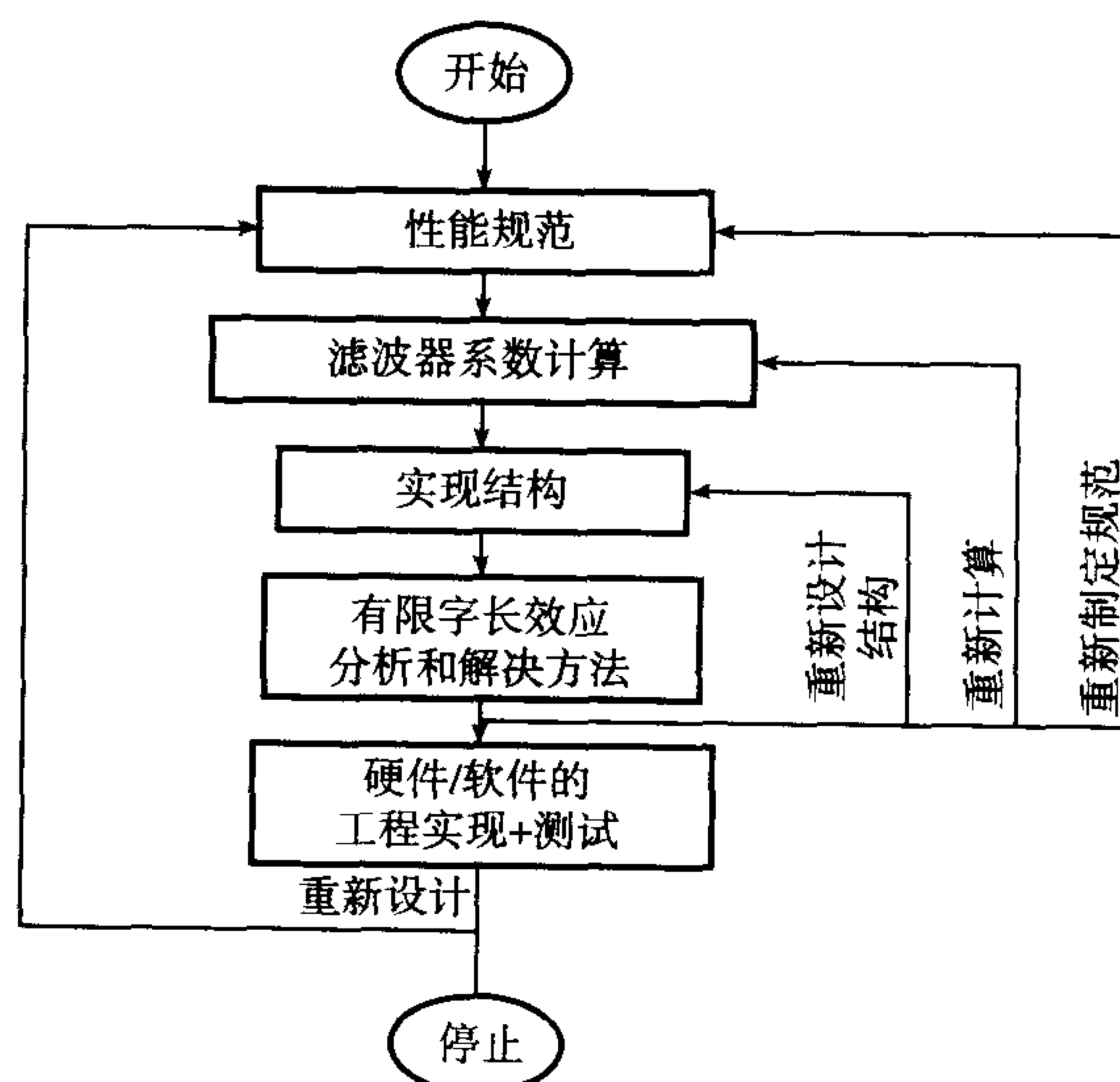


图7.2 数字滤波器设计步骤总结

7.3 FIR滤波器规范

在第6章我们详细讨论了滤波器规范，这里我们仅涉及其中与FIR滤波器有关的方面。本章的几个例子也将对滤波器规范的各方面进行说明。

对于相位响应，我们只需阐述是要求正对称的还是负对称的（假设线性相位）。FIR滤波器的幅度－频率响应常以允许设计方案（tolerance scheme）的形式来刻画。图7.3显示了低通滤波器的一种设计方案。对于其他的频率选择滤波器，类似的方案可以很容易地画出来。参考7.3图，下面是一些感兴趣的参数：

δ_p	峰值通带偏差（或波纹）
δ_s	阻带偏差
f_p	通带边缘频率
f_s	阻带边缘频率
F_s	抽样频率

实际上像图中那样用分贝表示δ_p和δ_s更加方便，f_s与f_p之差给出了滤波器的过渡带宽。另一个重要的参数是滤波器长度N，它定义为给出的滤波器系数的数目。在多数情况下，这些参数完全定义了FIR滤波器的频率响应。

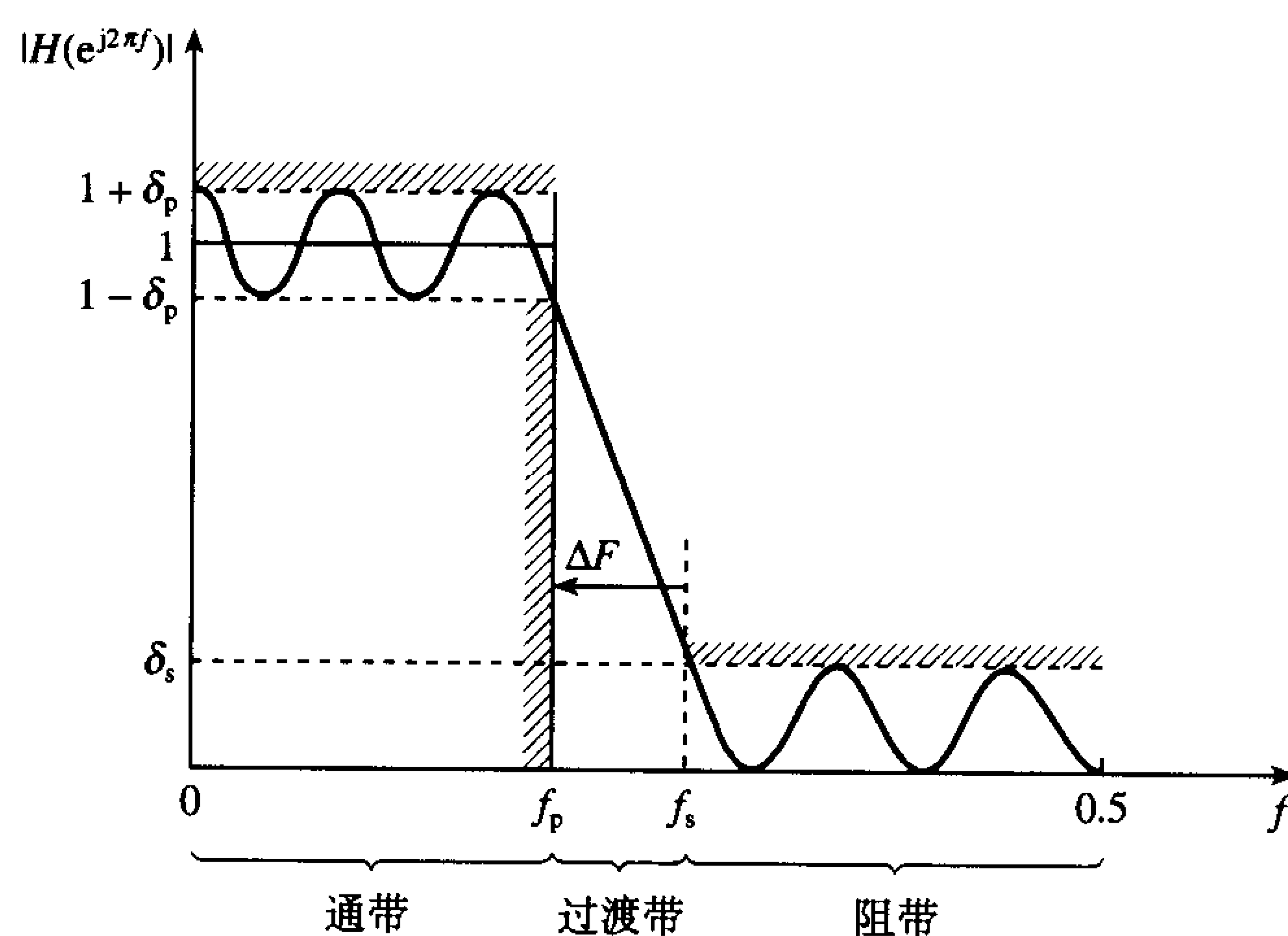


图7.3 低通滤波器的幅度频率响应规范。通带和阻带偏差常常用分贝表示：通带偏差，$20\log(1+\delta_p)$ dB；阻带偏差，$-20\log(\delta_s)$ dB

其他可能感兴趣的规范包括我们可接受的最大滤波器系数的数目（在一些特殊的应用中，这可能是我们必须接受的，比如我们希望一定的运算速度）。我们还没有选择上述参数的较好方法，所以只有通过反复试验来选择参数。

例7.2 振幅规范实例 降低生理噪声要求低通数字滤波器，滤波器应满足下面的技术规范：

通带的边缘频率	10 Hz
阻带的边缘频率	< 20 Hz
阻带衰减	>30 dB
通带波纹	< 0.026 dB
抽样频率	256 Hz

此应用的重要要求是(i) 在信号带内滤波器引入的失真尽可能小，(ii) 滤波器的长度应尽可能的小且不要超过37。

7.4　FIR 滤波器系数的计算方法

FIR 滤波器由下列方程来刻画：

$$y(m) = \sum_{n=0}^{N-1} h(n)x(m-n)$$

$$H(z) = \sum_{n=0}^{N-1} h(n)z^{-n}$$

大多数FIR系数的计算方法中，惟一目标是求$h(n)$的值，使得导出的滤波器满足设计规范，例如幅度－频率响应和吞吐率的要求。求$h(n)$的有效方法有几种，然而最常用的方法是窗口方法、最佳方法及频率抽样方法。这三种方法均可导出线性相位 FIR 滤波器。

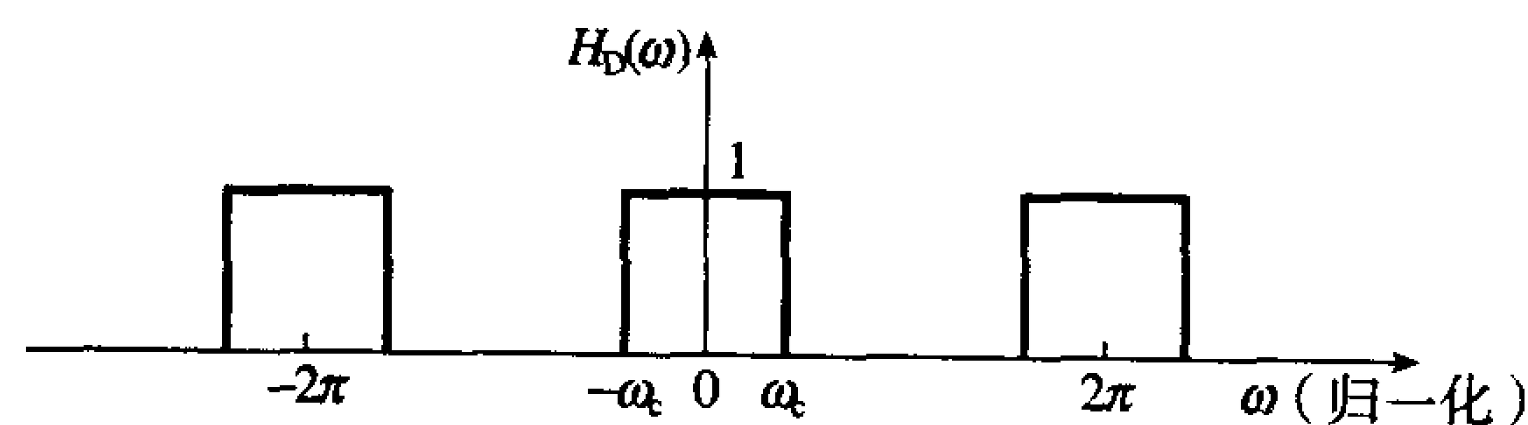
7.5　窗口方法

在这种方法中，利用了这样一个事实，滤波器的频率响应$H_D(\omega)$和相应冲激响应$h_D(n)$通过傅里叶反变换联系起来：

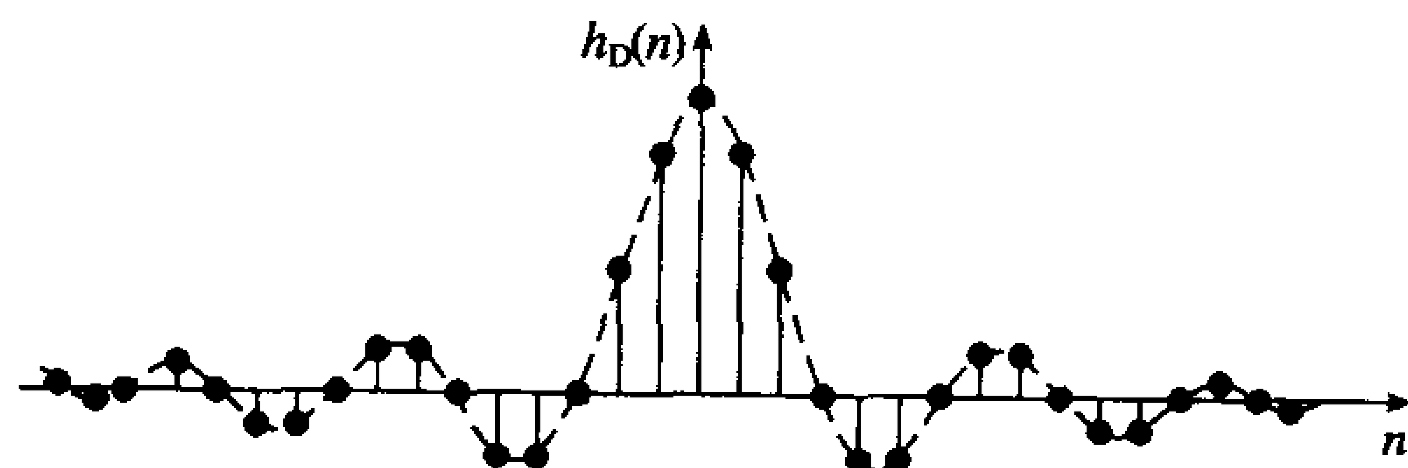
$$h_D(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_D(\omega)e^{j\omega n}d\omega \tag{7.5}$$

下标 D 用来区别理想的和实际的冲激响应。不久读者就会明白为什么需要区分这些。如果我们已知$H_D(\omega)$，通过计算7.5式的傅里叶反变换就可以求得$h_D(n)$。作为一个例子，假定我们要设计一个低通滤波器，我们应该从图7.4(a)所示的理想低通响应开始，其中ω_c是截止频率，且频率刻度被归一化为$T=1$。通过令响应从$-\omega_c$到ω_c，我们简化积分运算。这样，冲激响应为

$$\begin{aligned} h_D(n) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} 1\times e^{j\omega n}d\omega = \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} e^{j\omega n}d\omega \\ &= \frac{2f_c\sin(n\omega_c)}{n\omega_c}, \quad n\neq 0, -\infty \leqslant n \leqslant \infty \\ &= 2f_c, \qquad n=0\text{（使用L'Hôpital规则）} \end{aligned} \tag{7.6}$$



(a) 低通滤波器的理想频率响应



(b) 理想低通滤波器的冲激响应

图7.4　低通滤波器的理想频率响应和理想低通滤波器的冲激响应

理想高通、带通和带阻滤波器的冲激响应从 7.6 式的低通情况下得到，并且总结在表 7.2 中。低通滤波器的冲激响应画在图 7.4b 中，从图中我们看出 $h_D(n)$关于 $n = 0$ 对称（即 $h_D(n) = h_D(-n)$），因此滤波器具有线性相位响应。根据这一简单方法，几个实用问题显而易见，其中尤为重要的是虽然 $h_D(n)$随着与 $n = 0$ 的距离的增加而递减，然而理论上它一直延续到 $n = \pm\infty$，这样得出的滤波器不是 FIR 滤波器。

表 7.2　标准频率选择性滤波器的理想冲激响应总结

滤波器类型	理想冲激响应，$h_D(n)$	
	$h_D(n), n \neq 0$	$h_D(0)$
低通	$2f_c\dfrac{\sin(n\omega_c)}{n\omega_c}$	$2f_c$
高通	$-2f_c\dfrac{\sin(n\omega_c)}{n\omega_c}$	$1-2f_c$
带通	$2f_2\dfrac{\sin(n\omega_2)}{n\omega_2} - 2f_1\dfrac{\sin(n\omega_1)}{n\omega_1}$	$2(f_2-f_1)$
带阻	$2f_1\dfrac{\sin(n\omega_1)}{n\omega_1} - 2f_2\dfrac{\sin(n\omega_2)}{n\omega_2}$	$1-2(f_2-f_1)$

f_c、f_1 和f_2 是归一化的通带或带阻边缘频率；N 是滤波器长度。

一个明显的解决方法是当 n 大于 M 时截断理想冲激响应（设 $h_D(n) = 0$），然而这样会引入不希望的波纹和过冲量，这就是所谓的吉布斯现象（Gibb's phenomenon）。图 7.5 说明了丢弃系数对滤波器响应的影响。保留的系数越多滤波器频谱越接近理想响应（参见图 7.5(b)和图 7.5(c)）。上述的直接截断 $h_D(n)$等价于与具有如下矩形窗形式的理想冲激响应相乘：

$$
\begin{aligned}
w(n) &= 1, \quad |n| = 0, 1, \ldots, (M-1)/2 \\
&= 0, \quad 其他
\end{aligned}
$$

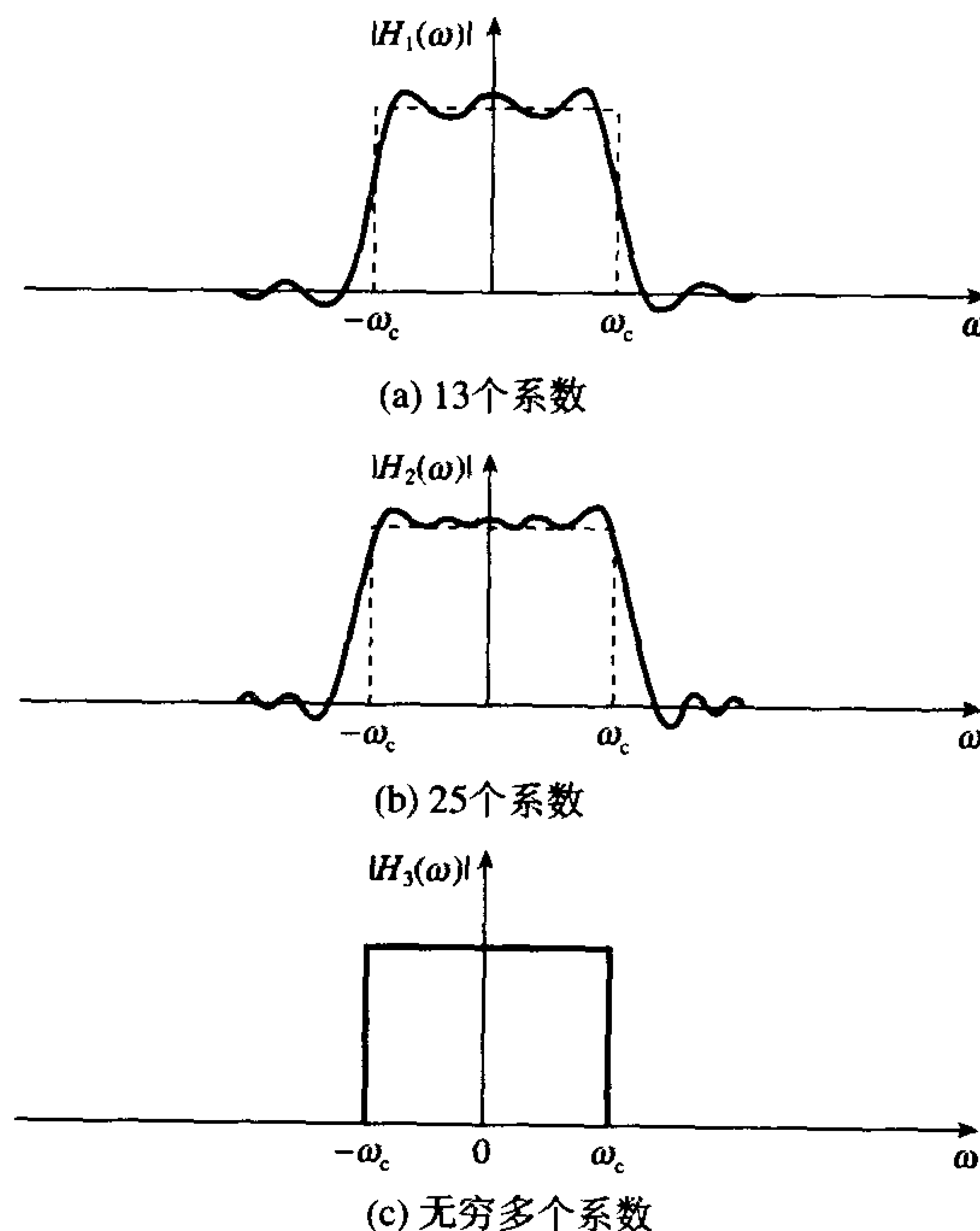


图 7.5　将理想冲激响应截断成(a)、(b)、(c)后对频率响应的影响

在频域这等价于 $H_D(\omega)$与 $W(\omega)$的卷积，其中 $W(\omega)$是 $w(n)$的傅里叶变换。当 $W(\omega)$形式为典型的 $(\sin x)/x$ 时，$h_D(n)$的截断导致频率响应的过冲和波纹。

一个实际的方法是选择合适的窗函数 $w(n)$与理想冲激响应 $h_D(n)$相乘，$w(n)$的持续时间是有限的。这种方法最终导致冲激响应平稳地衰减到零，这个过程如图7.6所示。图7.6(a)给出了理想频率响应及对应的冲激响应，图7.6(b)显示的是有限持续时间窗函数和它的频谱，图7.6(c)显示了 $h_D(n)$与 $w(n)$相乘得到的 $h(n)$，对应的频率响应说明波纹和过冲量（直接截断特性）减少了。而过渡带宽要比矩形情况下的宽。滤波器的过渡带宽取决于窗的主波瓣的宽度，旁瓣在通带和阻带中产生了波纹。

图7.6 如何用窗口方法求滤波器系数 $h(n)$的说明图

7.5.1 常用的窗函数

常见的几个窗函数中，最广泛使用的窗函数之一是哈明（Hamming）窗，哈明窗定义如下：

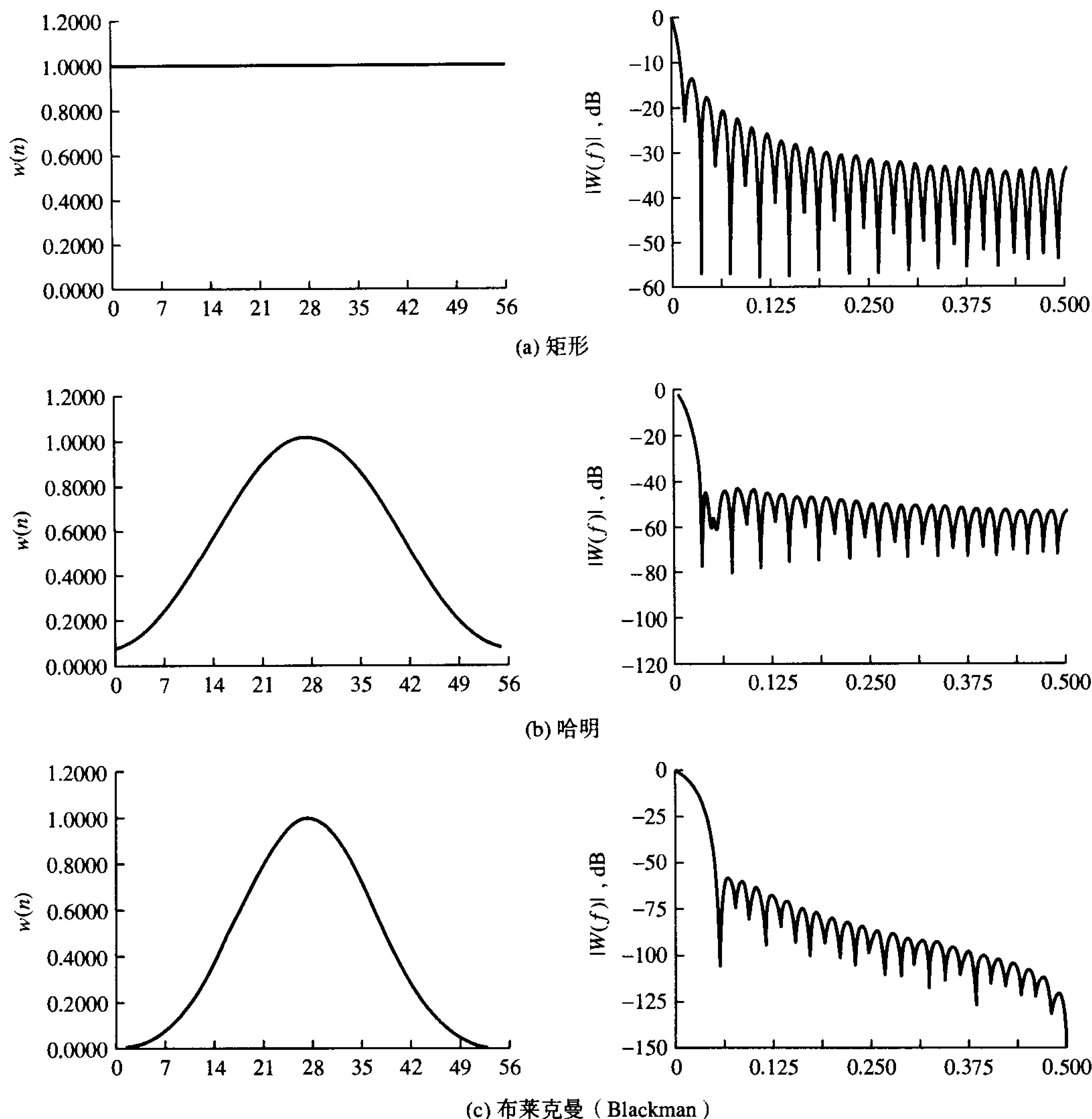
$$w(n) = 0.54 + 0.46\cos(2\pi n/N)\begin{cases}-(N-2)/2 \leqslant n \leqslant (N-1)/2 & (N\text{为奇数})\\ -N/2 \leqslant n \leqslant N/2 & (N\text{为偶数})\end{cases}$$
$$= 0 \quad \text{其他} \tag{7.7}$$

在图7.7中与矩形窗比较了时域与频域特征。在时域哈明窗函数向两边平缓地递减为零，在频域主瓣的幅度要比矩形窗宽（约2倍），但哈明窗旁瓣的幅值与主瓣相比要小大约40分贝，而矩形窗旁瓣的幅值与主瓣相比仅小14分贝。这意味着与矩形窗相比，哈明窗得出的滤波器具有较宽的过渡带宽（因为主瓣宽），但是有较高的阻带衰减（由于旁瓣电平较小）。

用哈明窗设计的滤波器的过渡带宽（从通带到阻带）和滤波器长度之间的关系为

$$\Delta f = 3.3/N \tag{7.8}$$

其中N为滤波器长度，Δf为归一化的过渡带宽。哈明窗可能的最大阻带衰减大致为 53 dB，最小峰值通带波纹大约为 0.0194 dB。



(a) 矩形

(b) 哈明

(c) 布莱克曼（Blackman）

图 7.7　常用窗函数的时域与频域特征的比较

表 7.3 总结了一些最流行的窗函数的有关特征。我们注意到前四个窗函数有固定的特性，如过渡带宽和阻带衰减。因此它们的使用约束了滤波器的设计者。我们还注意到由窗口方法设计的滤波器有相同的通带波纹和阻带波纹，即 $\delta_p = \delta_s$（参见图 7.3）。实际上这种约束可能导致滤波器通带波纹不必要的小。

凯塞窗（Kaiser）函数结合波纹控制参数β在解决上述问题时有些用处，波纹控制参数β允许设计者在过渡带宽与波纹之间进行折中。凯塞窗为

$$w(n) = I_0\left\{\beta\left[1-\left(\frac{2n}{N-1}\right)^2\right]^{1/2}\right\}\Big/ I_0(\beta) \quad -(N-1)/2 \leqslant n \leqslant (N-1)/2$$
$$= 0 \qquad \text{其他} \tag{7.9}$$

其中$I_0(x)$是第一类零阶修正贝塞尔（Bessel）函数，β在时域控制窗函数在边缘变尖的方式。$I_0(x)$通常用下列幂级数的形式计算（Rabiner and Gold, 1975）：

$$I_0(x) = 1 + \sum_{k=1}^{L}\left[\frac{(x/2)^k}{k!}\right]^2$$

其中通常$L < 25$。由凯塞（Rabiner and Gold, 1975）提出的一种算法给出了这一方程的有效实现方法。

当$\beta = 0$时，凯塞窗对应于矩形窗；当$\beta = 5.44$时，得出的窗非常类似于哈明窗，尽管不完全相同。β的值由要求的阻带衰减来确定，且可由下列经验关系来估算：

$$\beta = 0 \qquad \text{如果}\ A \leqslant 21\,\text{dB} \tag{7.10a}$$

$$\beta = 0.5842(A-21)^{0.4} + 0.078\,86(A-21) \qquad \text{如果}\ 21\,\text{dB} < A < 50\,\text{dB} \tag{7.10b}$$

$$\beta = 0.1102(A-8.7) \qquad \text{如果}\ A \geqslant 50\,\text{dB} \tag{7.10c}$$

其中$A = -20\log_{10}(\delta)$是阻带衰减，$\delta = \min(\delta_p, \delta_s)$，而通带和阻带波纹几乎相等，$\delta_p$是期望的通带波纹，$\delta_s$是期望的阻带波纹。滤波器系数的个数$N$为

$$N \geqslant \frac{A - 7.95}{14.36\Delta f} \tag{7.11}$$

其中Δf是归一化的过渡带宽，β和N的值用来计算凯塞窗$w(n)$的系数。

表 7.3 常用窗函数重要特征总结

窗函数名	过渡带宽（Hz）（归一化）	通带波纹（dB）	相对于旁瓣的主瓣（dB）	阻带衰减（dB）（最大值）	窗函数 $w(n)$, $\lvert n\rvert \leqslant (N-1)/2$
矩形	$0.9/N$	0.7416	13	21	1
汉宁（Hanning）	$3.1/N$	0.0546	31	44	$0.5 + 0.5\cos\left(\frac{2\pi n}{N}\right)$
哈明	$3.3/N$	0.0194	41	53	$0.54 + 0.46\cos\left(\frac{2\pi n}{N}\right)$
布莱克曼	$5.5/N$	0.0017	57	75	$0.42 + 0.5\cos\left(\frac{2\pi n}{N-1}\right) + 0.08\cos\left(\frac{4\pi n}{N-1}\right)$
	$2.93/N(\beta = 4.54)$	0.0274		50	$\frac{I_0(\beta\{1-[2n/(N-1)]^2\}^{1/2})}{I_0(\beta)}$
凯塞	$4.32/N(\beta = 6.76)$	0.002 75		70	
	$5.71/N(\beta = 8.96)$	0.000 275		90	

7.5.2 计算 FIR 滤波器系数的窗口方法总结

- **第一步** 指定理想的或期望的滤波器频率响应$H_D(\omega)$；
- **第二步** 通过计算傅里叶反变换 7.6b 式求期望的滤波器的冲激响应，对于标准的频率选择滤波器，$h_D(n)$的表达式总结在表 7.2 中；
- **第三步** 选择一个满足通带或衰减指标的窗函数，然后利用滤波器长度与过渡带宽Δf（表示为抽样频率的分数）之间的关系确定滤波器系数的数目；
- **第四步** 对于选取的窗函数求$w(n)$的值，并且将$h_D(n)$与$w(n)$相乘求得实际的 FIR 系数$h(n)$：

$$h(n) = h_D(n)w(n) \tag{7.12}$$

很显然，窗口方法是很简单的，包含的计算量较小。实际上，通过计算器就可以求出系数，然而，基于PC的计算$h(n)$的程序可在指导手册的CD上找到，应该说得出的滤波器不是最佳的。在许多情况下，通过其他方法可以求得系数少一些的滤波器。

例7.3　利用窗口法求满足下列技术规范的FIR低通滤波器系数。

通带边缘频率	1.5 kHz
过渡带宽	0.5 kHz
阻带衰减	> 50 dB
抽样频率	8 kHz

解：

从表7.2，对于低通滤波器，我们选择$h_D(n)$为

$$h_D(n) = 2f_c\frac{\sin(n\omega_c)}{n\omega_c} \quad n \neq 0$$

$$h_D(n) = 2f_c \quad n = 0$$

表7.3表明哈明窗、凯塞窗或布莱克曼窗都满足阻带衰减要求。为了简化起见，我们使用哈明窗。令$\Delta f = 0.5/8 = 0.0625$。由$N = 3.3/\Delta f = 3.3/0.0625 = 52.8$，设$N = 53$，则滤波器系数可从下面的式子得出：

$$h_D(n)w(n) \qquad -26 \leqslant n \leqslant 26$$

其中

$$h_D(n) = \frac{2f_c\sin(n\omega_c)}{n\omega_c} \qquad n \neq 0$$

$$h_D(n) = 2f_c \qquad n = 0$$

$$w(n) = 0.54 + 0.46\cos(2\pi n/53) \quad -26 \leqslant n \leqslant 26$$

因为对滤波器响应加窗的拖尾效应，得到的滤波器的截止频率将与技术规范中给出的不同。为了说明这个问题，我们使用过渡带宽中间点的f_c：

$$f_c' = f_c + \Delta f/2 = (1.5 + 0.25)\ \text{kHz} = 1.75\ \text{kHz} \rightarrow 1.75/8 = 0.218\,75$$

注意到$h(n)$是对称的，因此只需要计算$h(0), h(1),\ldots, h(26)$，然后利用对称性质得到其他系数：

$n = 0$:

$$h_D(0) = 2f_c = 2 \times 0.218\,75 = 0.4375$$

$$w(0) = 0.54 + 0.46\cos(0) = 1$$

$$h(0) = h_D(0)w(0) = 0.4375$$

$n = 1$:

$$h_D(1) = \frac{2 \times 0.218\,75}{2\pi \times 0.218\,75}\sin(2\pi \times 0.218\,75)$$

$$= \frac{\sin(360^\circ \times 0.218\,75)}{\pi} = 0.312\,19$$

$$w(1) = 0.54 + 0.46\cos(2\pi/53)$$

$$= 0.54 + 0.46\cos(360^\circ/53) = 0.996\,77$$

$$h(1) = h(-1) = h_D(1)w(1) = 0.311\,18$$

$$n = 2: \quad h_D(2) = \frac{2 \times 0.218\,75}{2 \times 2\pi \times 0.218\,75} \sin(2 \times 2\pi \times 0.218\,75)$$

$$= \frac{\sin(157.5°)}{2\pi} = 0.060\,13$$

$$w(2) = 0.54 + 0.46\cos(2\pi \times 2/53)$$

$$= 0.54 + 0.46\cos(720°/53) = 0.987\,13$$

$$h(2) = h(-2) = h_D(2)w(2) = 0.060\,12$$

$$\vdots \quad \vdots \quad \vdots \quad \vdots$$

$$n = 26: \quad h_D(26) = \frac{2 \times 0.218\,75}{26 \times 2\pi \times 0.218\,75} \sin(26 \times 2\pi \times 0.218\,75)$$

$$= -0.011\,31$$

$$w(26) = 0.54 + 0.46\cos(2\pi \times 26/53)$$

$$= 0.54 + 0.46\cos(9360°/53) = 0.080\,81$$

$$h(26) = h(-26) = h_D(26)w(26) = -0.000\,914$$

注意到滤波器系数下标是从−26到26。为了使滤波器是因果系统（实现所必须的），我们将每个下标加26，这样滤波器系数下标就从0开始。表7.4列出了下标调整后的滤波器系数。滤波器频谱（没画出）说明滤波器系数满足技术规范。

表7.4 例7.3（$N = 53$，哈明窗，$f_c = 1750$ Hz）的FIR系数

h[0] =	−9.1399895e−04	= h[52]
h[1] =	2.1673690e−04	= h[51]
h[2] =	1.3270280e−03	= h[50]
h[3] =	3.2138355e−04	= h[49]
h[4] =	−1.9238177e−03	= h[48]
h[5] =	−1.4683633e−03	= h[47]
h[6] =	2.3627318e−03	= h[46]
h[7] =	3.4846558e−03	= h[45]
h[8] =	−1.9925839e−03	= h[44]
h[9] =	−6.2837232e−03	= h[43]
h[10] =	4.5320247e−09	= h[42]
h[11] =	9.2669460e−03	= h[41]
h[12] =	4.3430566e−03	= h[40]
h[13] =	−1.1271299e−02	= h[39]
h[14] =	−1.1402453e−02	= h[38]
h[15] =	1.0630714e−02	= h[37]
h[16] =	2.0964392e−02	= h[36]
h[17] =	−5.2583216e−03	= h[35]
h[18] =	−3.2156086e−02	= h[34]
h[19] =	−7.5449714e−03	= h[33]
h[20] =	4.3546153e−02	= h[32]
h[21] =	3.2593190e−02	= h[31]
h[22] =	−5.3413653e−02	= h[30]
h[23] =	−8.5662029e−02	= h[29]
h[24] =	6.0122145e−02	= h[28]
h[25] =	3.1118568e−01	= h[27]
h[26] =	4.3750000e−01	= h[26]

例7.4　FIR滤波器为了满足下面的技术规范，存在一定的要求：

通带	150 ~ 250 Hz
过渡带宽	50 Hz
通带波纹	0.1 dB
阻带衰减	60 dB
抽样频率	1 kHz

使用窗口方法计算出滤波器系数和频谱。

解：

由上述技术规范，通带和阻带波纹为

$$20\log(1+\delta_p)=0.1\text{ dB},\quad 得\quad \delta_p=0.0115$$

$$-20\log(\delta_s)=60\text{ dB},\quad 得\quad \delta_s=0.001$$

因此，

$$\delta=\min(\delta_p,\delta_s)=0.001$$

仅凯塞窗和布莱克曼窗满足衰减要求。对于凯塞窗，滤波器系数的数目为

$$N\geqslant\frac{A-7.95}{14.36\Delta F}=\frac{60-7.95}{14.36(50/1000)}=72.49$$

令$N=73$。波纹参数为

$$\beta=0.1102(60-8.7)=5.65$$

且$N=73$, $\beta=5.65$，程序window.c（参见附录）用来计算$w(n)$的值，以及理想冲激响应$h_D(n)$和滤波器系数。考虑到窗函数的拖尾效应，在计算理想冲激响应时，截止频率采用$f_{c1}-\Delta f/2$和$f_{c2}+\Delta f/2$，即$f_{c1}=125$ Hz，$f_{c2}=275$ Hz。表7.5给出了滤波器系数，图7.8给出了滤波器频谱。对于布莱克曼窗，滤波器系数数目的估计可按如下方式求得：

$$N=5.5/\Delta f=5.5/(50/1000)\approx 110$$

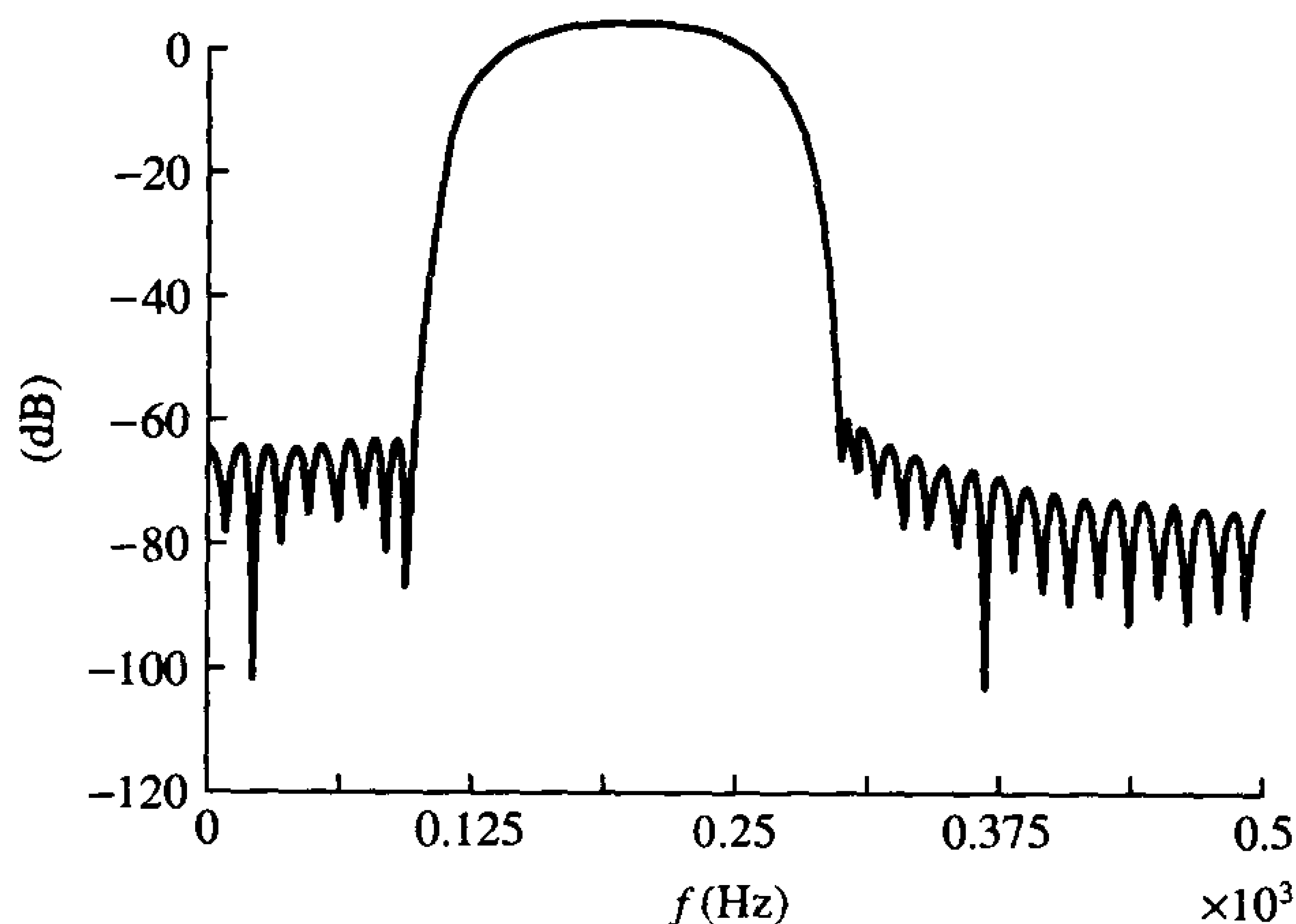


图7.8　滤波器的频谱（例7.4）

表 7.5 凯塞滤波器系数（例 7.4）

h[0] =	−1.0627330e−04	= h[72]
h[1] =	−3.9118142e−04	= h[71]
h[2] =	−7.5561629e−05	= h[70]
h[3] =	−1.3695577e−04	= h[69]
h[4] =	−6.8122013e−04	= h[68]
h[5] =	5.0929290e−04	= h[67]
h[6] =	2.3413494e−03	= h[66]
h[7] =	8.0280013e−04	= h[65]
h[8] =	−1.7031635e−04	= h[64]
h[9] =	−5.5034956e−04	= h[63]
h[10] =	−4.9912488e−04	= h[62]
h[11] =	−4.4036355e−03	= h[61]
h[12] =	−2.1639856e−03	= h[60]
h[13] =	6.9094151e−03	= h[59]
h[14] =	6.6067599e−03	= h[58]
h[15] =	−1.6445200e−03	= h[57]
h[16] =	4.5229777e−09	= h[56]
h[17] =	2.1890066e−03	= h[55]
h[18] =	−1.1720511e−02	= h[54]
h[19] =	−1.6377726e−02	= h[53]
h[20] =	6.8804519e−03	= h[52]
h[21] =	1.8882837e−02	= h[51]
h[22] =	2.9068601e−03	= h[50]
h[23] =	4.3925286e−03	= h[49]
h[24] =	1.8839744e−02	= h[48]
h[25] =	−1.2481155e−02	= h[47]
h[26] =	−5.2063428e−02	= h[46]
h[27] =	−1.6557375e−02	= h[45]
h[28] =	3.3298453e−02	= h[44]
h[29] =	1.0439025e−02	= h[43]
h[30] =	9.4320244e−03	= h[42]
h[31] =	8.5673629e−02	= h[41]
h[32] =	4.5314758e−02	= h[40]
h[33] =	−1.6657147e−01	= h[39]
h[34] =	−2.0669512e−01	= h[38]
h[35] =	8.9135544e−02	= h[37]
h[36] =	3.0000000e−01	= h[36]

由于空间有限，布莱克曼窗滤波器系数表在这里没有给出。为了满足相同规范，对于需要的系数数目，凯塞窗要比布莱克曼窗更有效。总之，在滤波器系数数目上凯塞窗要比其他窗更有效。

例 7.5 利用凯塞窗求满足下面幅度响应规范的线性相位 FIR 滤波器系数：

阻带衰减	40 dB
通带波纹	0.01 dB
过渡带宽	500 Hz
抽样频率	10 kHz
理想截止频率	1200 Hz

解：

由技术规范，

$$20\log(1+\delta_p)=0.01\ \text{dB},\quad 得\quad \delta_p=0.001\,15$$

$$-20\log(\delta_s)=40\ \text{dB},\quad 得\quad \delta_s=0.01$$

由于在窗口方法中通带波纹和阻带波纹相等（它们无法单独指定），我们取波纹最小者：

$$\delta=\delta_s=\delta_p=0.001\,15$$

这也就意味着在 $-20\log(0.001\,15)=58.8$ dB 的情况下，阻带衰减要比实际要求的大。

根据 7.11 式，要求的滤波器系数数目为

$$N=\frac{A-7.95}{14.36\Delta f}=\frac{58.8-7.95}{14.36(500/10\,000)}\approx 71$$

而如果要求的衰减指标为 40 dB，那么 N 应该是 45。因此，在窗口方法中 δ_p 等于 δ_s 的要求导致滤波器系数的数目高于必需的数目。

由 7.10 式，波纹参数为

$$\beta=0.5842(58.8-21)^{0.4}+0.078\,86(58.8-21)=5.48$$

FIR 系数由 $h(n)=h_D(n)w(n)$求得，其中由表 7.2，

$$h_D(n)=2f_c\frac{\sin(n\omega_c)}{n\omega_c}\qquad n\neq 0$$

$$h_D(n)=2f_c\qquad n=0$$

$w(n)$由 7.9 式给出。

像前面解释的那样，考虑到拖尾效应，计算 $h(n)$用到的截止频率f_c与技术规范中给出的f_c不同。我们选择过渡频带中间的f_c值：$f_c'=1200+\Delta f/2=1450$ Hz。

使用计算机程序 window.c（参见附录）时采用下面的滤波器参数：

截止频率	1450 Hz
波纹参数 β	5.48
滤波器系数数目	71
抽样频率	10 kHz

滤波器系数在表 7.6 中给出，滤波器频谱在图 7.9 给出。

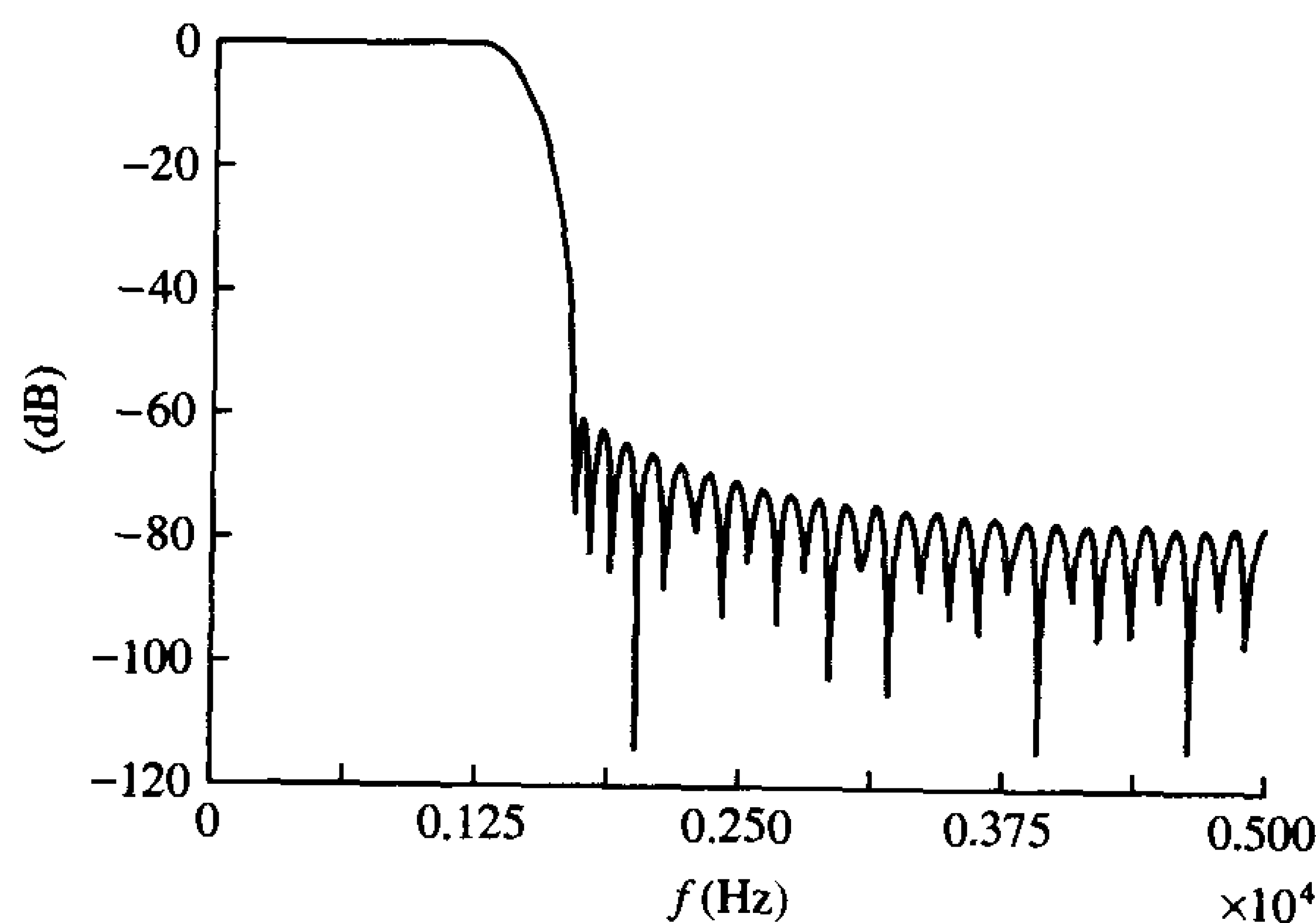


图 7.9　凯塞窗下的滤波器频谱（例 7.5）

表 7.6 使用凯塞窗的滤波器系数（例 7.5）

h[0] =	9.8470163e-05	= h[70]
h[1] =	-1.3972411e-04	= h[69]
h[2] =	-4.5442489e-04	= h[68]
h[3] =	-4.8756977e-04	= h[67]
h[4] =	2.6173965e-05	= h[66]
h[5] =	8.6653647e-04	= h[65]
h[6] =	1.2967984e-03	= h[64]
h[7] =	6.1688894e-04	= h[63]
h[8] =	-1.0445340e-03	= h[62]
h[9] =	-2.4646644e-03	= h[61]
h[10] =	-2.1059775e-03	= h[60]
h[11] =	4.4371801e-04	= h[59]
h[12] =	3.5954580e-03	= h[58]
h[13] =	4.5526695e-03	= h[57]
h[14] =	1.5922295e-03	= h[56]
h[15] =	-3.8904820e-03	= h[55]
h[16] =	-7.6398162e-03	= h[54]
h[17] =	-5.6061945e-03	= h[53]
h[18] =	2.2010888e-03	= h[52]
h[19] =	1.0450148e-02	= h[51]
h[20] =	1.1760002e-02	= h[50]
h[21] =	2.8239875e-03	= h[49]
h[22] =	-1.1380549e-02	= h[48]
h[23] =	-1.9631856e-02	= h[47]
h[24] =	-1.2665935e-02	= h[46]
h[25] =	8.0061777e-03	= h[45]
h[26] =	2.8182781e-02	= h[44]
h[27] =	2.9474031e-02	= h[43]
h[28] =	3.8724896e-03	= h[42]
h[29] =	-3.5942288e-02	= h[41]
h[30] =	-5.9766794e-02	= h[40]
h[31] =	-3.7113570e-02	= h[39]
h[32] =	4.1378026e-02	= h[38]
h[33] =	1.5291289e-01	= h[37]
h[34] =	2.5100632e-01	= h[36]
h[35] =	2.9000000e-01	= h[35]

7.5.3 窗口方法的优缺点

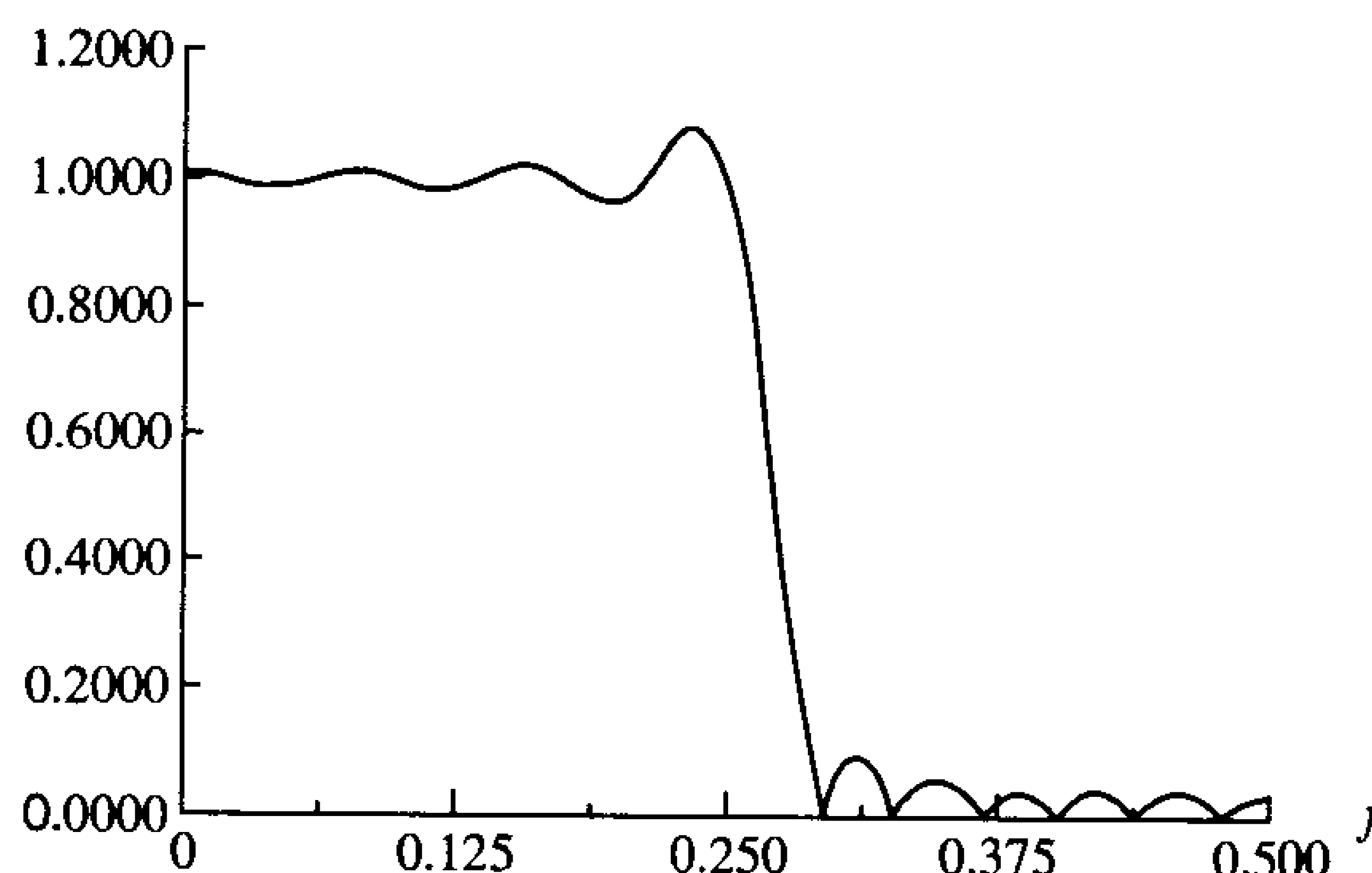
- 窗口方法重要的优点是其简单性：应用简单，理解简单。窗口方法需要的计算量最小，即使对于最复杂的凯塞窗也是如此。
- 窗口方法最大的缺陷是缺乏灵活性。由于峰值通带和阻带波纹近似相等，致使设计者最终因要么通带波纹太小、要么阻带衰减太大而中止设计过程。
- 由于窗函数频谱与期望响应之间的卷积效应，致使通带边缘频率和阻带边缘频率不能精确指定。
- 对于一个给定的窗口（凯塞窗口除外），无论 N 取多长，滤波器响应中的最大波纹幅度是固定的。因此，对于给定的窗口，它的阻带衰减也是固定的，滤波器设计者应根据给定的衰减指标确定合适的窗。
- 在某些应用中，对于 7.5 式用解析的方法求 $h_D(n)$ 来说，$H_D(\omega)$ 的表达式太复杂。在这种情况下，在应用窗函数之前，$h_D(n)$ 可通过频率抽样的方法求得（参见 7.7.1 节）。

7.6　最佳方法

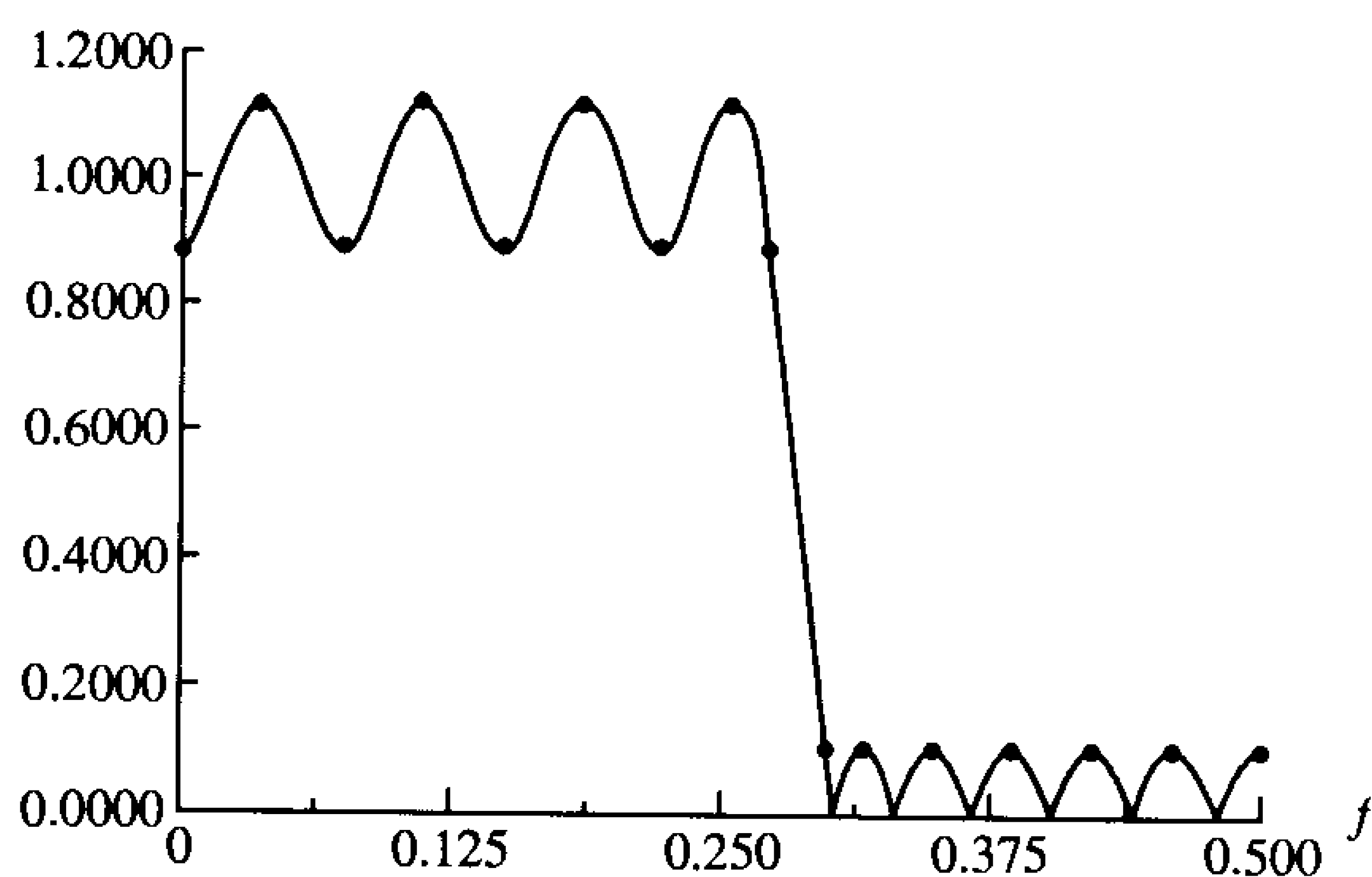
计算FIR滤波器系数的最佳方法是功能强大且非常灵活的，因为有优秀的设计程序并且容易应用。由于上述原因及其设计出的优秀的滤波器，最佳方法已成为许多FIR应用中的首选方法。下面讨论最佳方法的基本概念、设计程序及其应用，并用几个例子来说明该方法。

7.6.1　基本概念

很明显，在窗口方法中，在计算合适的滤波器系数的过程中固有的问题就是对期望的滤波器或理想频率响应做近似。由窗口方法设计的滤波器的峰值波纹出现在带沿频率附近，并且随着偏离带沿频率而递减（参见图7.10(a)）。如果波纹在通带和阻带上是均匀的，如图7.10(b)所示，我们就会得到期望滤波器频率响应的一个较好的近似值。



(a) 窗口方法设计的滤波器



(b) 最佳滤波器

图7.10　频率响应的比较。在(a)中，波纹在带沿附近最大，在(b)中，波纹在通带和阻带有同样的峰值（等波纹）

最佳方法是基于等通带和阻带波纹的概念。考虑一个图7.11所描述的低通滤波器频率响应。在通带内，实际的响应在$1-\delta_p$和$1+\delta_p$之间振荡；在阻带中，滤波器响应位于0到δ_s之间。理想滤波器和实际响应之间的差可以看作为一个误差函数：

$$E(\omega) = W(\omega)[H_D(\omega) - H(\omega)] \tag{7.13}$$

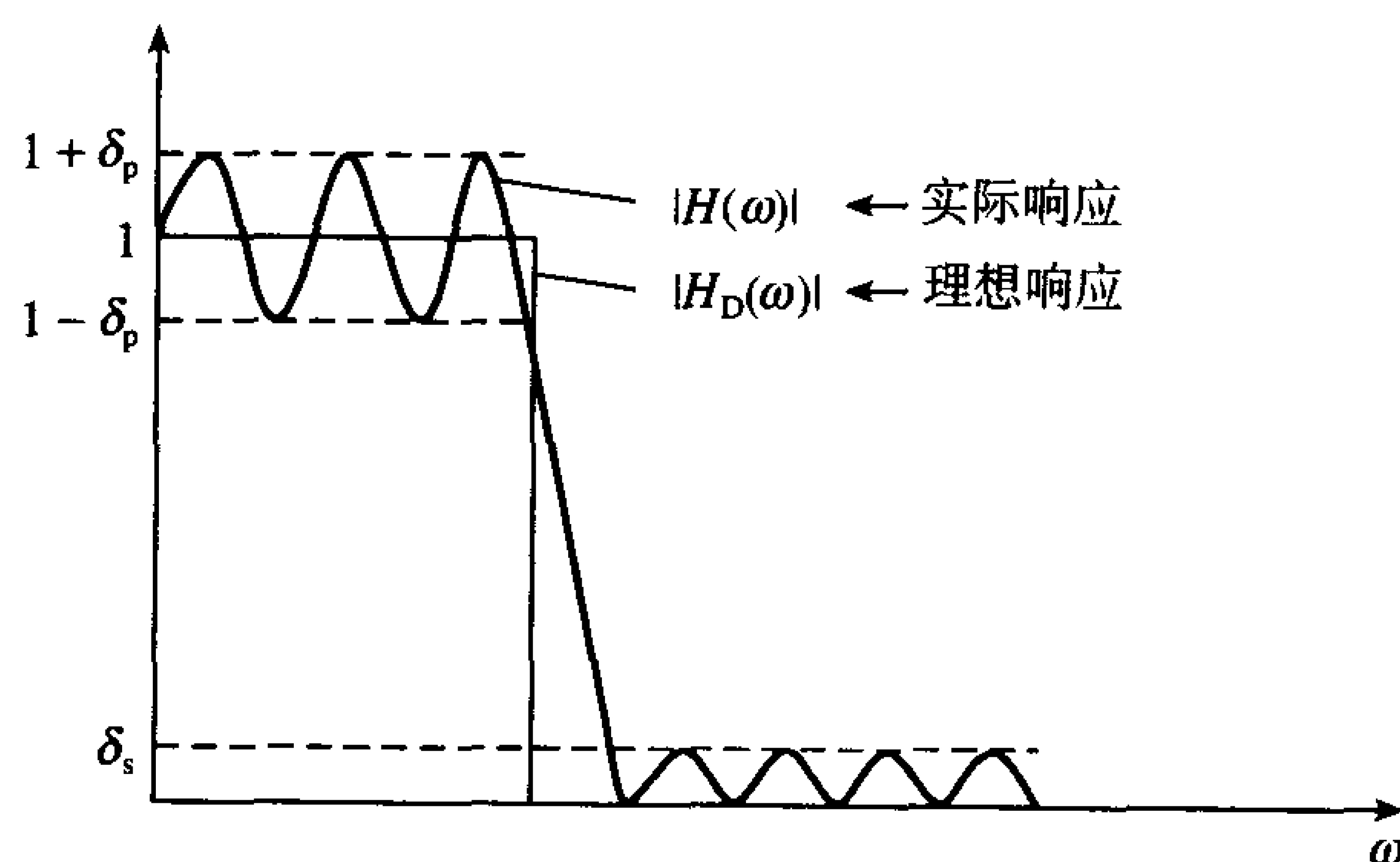
其中$H_D(\omega)$为理想的或期望的函数，$W(\omega)$是一个权函数，它允许在不同的频带之间定义近似的相对误差。在最佳方法中，目的就是要确定滤波器系数$h(n)$，使得最大加权值$|E(\omega)|$在通带和阻带内最小，数学上这可以表示为在通带和阻带内，

$$\min[\max|E(\omega)|]$$

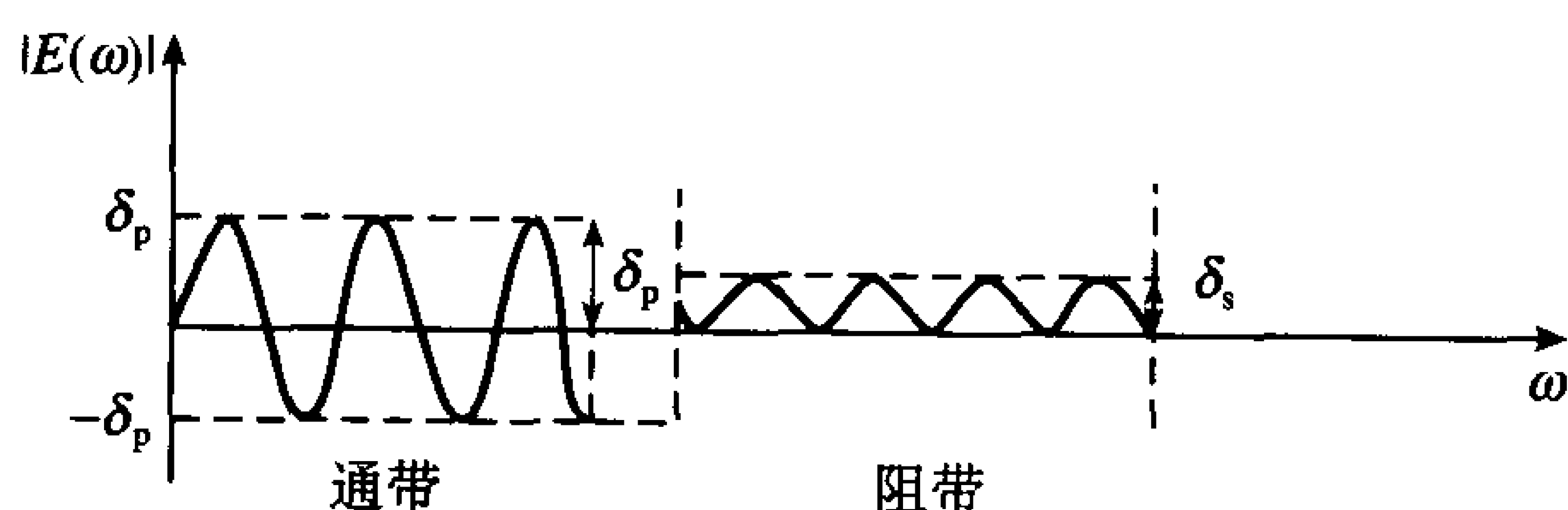
已经证明（例如参考 Rabiner and Gold, 1975），当 $|E(\omega)|$ 最小时，设计出的滤波器响应将是等通带和阻带波纹的，在两相等振幅电平之间信号有交替变化的波纹（参见图 7.10(b)）。最大值和最小值常称为极值，例如，对于线性相位低通滤波器，存在极值 $r+1$ 或者 $r+2$，其中 $r=(N+1)/2$（对类型 1 滤波器），或 $r=N/2$（对类型 2 滤波器），极值频率在图 7.10(b)中用小圆点表示。

对于给定的一组滤波器规范，除了带沿频率处之外（即 $f=f_p$ 和 $f=F_s/2$），极值频率的位置是先验未知的。因此，最佳方法的主要问题是求极值频率的位置。一种应用 Remez 交换算法求极值频率的强大技术已经出现(Rabiner and Gold, 1975; McClellan et al., 1973; Oppenheim and Schaffer, 1975)。一旦知道极值频率所在位置，计算出实际的频率响应——也就是滤波器的冲激响应将是一件简单的事情。对于给定的一组技术规范（通带边沿频率、N 以及通带波纹和阻带波纹之比），最佳方法包含下面几个关键步骤（参见图 7.12）：

- 使用 Remez 交换算法求一组最佳极值频率；
- 用极值频率确定频率响应；
- 求冲激响应系数。



(a) 最佳低通滤波器的频率响应



(b) 理想响应与实际响应之间的误差响应（$\delta_p=2\delta_s$）

图 7.11 低通滤波器频率响应

最佳方法的核心是第一步，该步使用迭代处理来确定滤波器的极值频率，该滤波器的幅度-频率响应满足最佳条件。这一步依赖于交替定理（alternation theorem），对于给定的 N，指定了可能存在的极值频率个数。

实现以上过程的 FORTRAN 程序是可用的，现在已被广泛地使用（McClellan et al., 1973）。等价的C语言程序在指导手册的CD上也可以找到。该程序支持各种频率选择性滤波器设计，包括低通、高通、带通和带阻以及差分器和希尔伯特变换器，它也具有计算用户指定的任意频率响应系数的能力。此外，最佳方法的详细内容在上面给出的参考文献中也可以找到，在附录中讨论了基于 MATLAB 的最佳方法的实现问题。

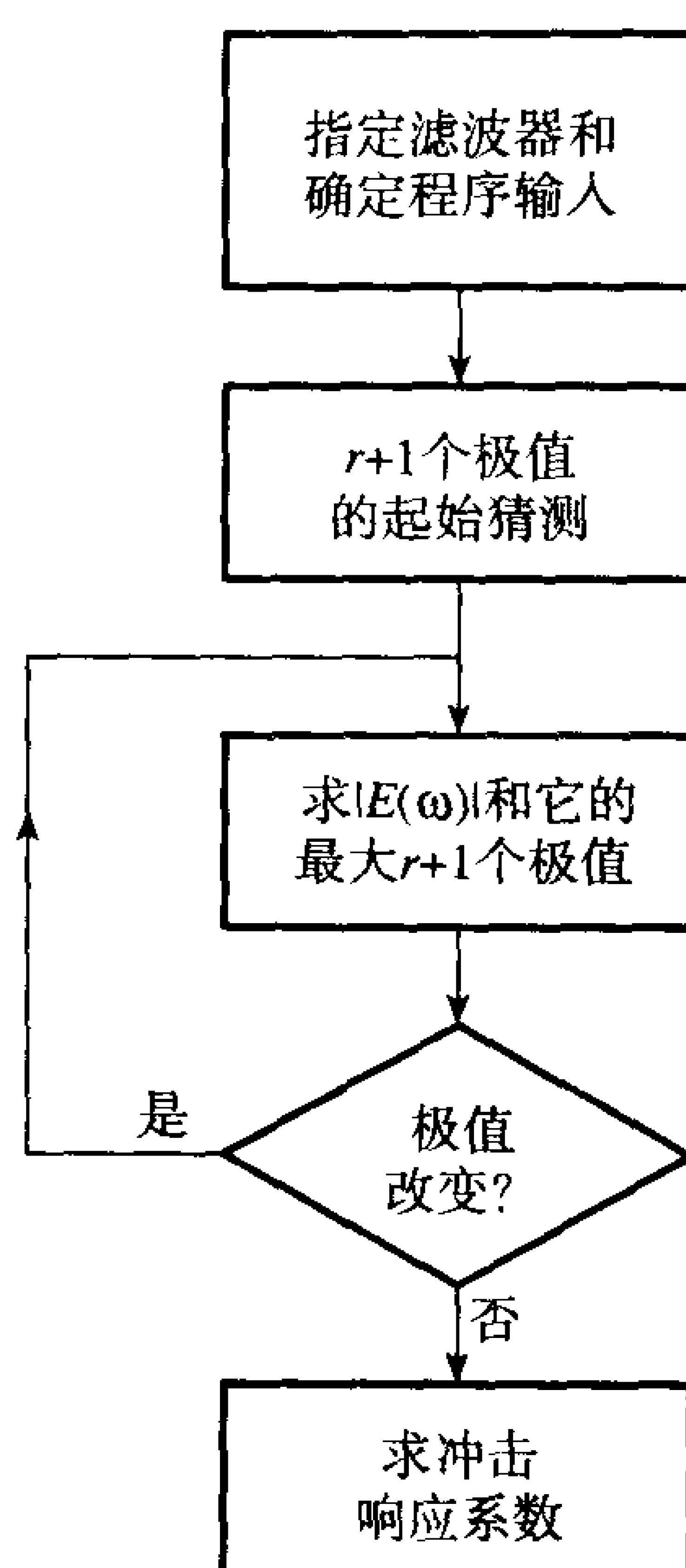


图 7.12　最佳方法的简化流程图

7.6.2　使用最佳程序所要求的参数

为了使用设计程序，用户必须提供一组描述滤波器的输入参数，这些参数包括：

N　滤波器系数数目，即滤波器长度，从下一节给定的关系中可以估计这个参数。

Jtype　这个参数指定了滤波器类型，滤波器有三种类型：Jtype = 1（多通带/阻带滤波器，包括低通、高通、带通和带阻滤波器），Jtype =2（指定差分器），Jtype = 3（指定希尔伯特变换器）。

$W(\omega)$　加权函数。这个参数规定了每一频带的相对重要性。实际上它允许在通带波纹和阻带衰减之间进行折中。对每一个频带规定一个权值。

Ngrid　这个参数指定栅格密度。在求极值频率的过程中，为了观察最佳条件是否满足，这个数是频率响应需要检查的频率点数（也就是在通带和阻带最大误差幅度 $|E(\omega)|$ 小这样一种最佳意义下）。Ngrid 的默认值为 16，在多数设计中取 16、32 或 64 比较合适。

Edge　它指定带沿频率(即滤波器的下沿和上沿频率)。所有频率必须用规一化的形式输入，第一带沿为0，最后一个为0.5(对应于抽样频率的一半)。支持最大10频带(通带或阻带)。

上述符号在近来的实现中不是必需的。事实上，现在设计实现的程序有比原先更加友好的用户界面。

7.6.3　估计滤波器长度 N 的关系

实际上滤波器系数数目是未知的，它的值可以由下面的经验公式估计出来。

7.6.3.1　低通滤波器（Herrman et al., 1973）

$$N \simeq \frac{D_\infty(\delta_p, \delta_s)}{\Delta F} - f(\delta_p, \delta_s)\Delta F + 1 \tag{7.14}$$

这里 ΔF 是用抽样频率归一化的过渡带宽，

$$D_\infty(\delta_p, \delta_s) = \log_{10}\delta_s[a_1(\log_{10}\delta_p)^2 + a_2\log_{10}\delta_p + a_3] + [a_4(\log_{10}\delta_p)^2 + a_5\log_{10}\delta_p + a_6]$$

$$f(\delta_p, \delta_s) = 11.012\,17 + 0.512\,44[\log_{10}\delta_p - \log_{10}\delta_s]$$

$$a_1 = 5.309\times10^{-3};\qquad a_2 = 7.114\times10^{-2}$$

$$a_3 = -4.761\times10^{-1};\qquad a_4 = -2.66\times10^{-3}$$

$$a_5 = -5.941\times10^{-1};\qquad a_6 = -4.278\times10^{-1}$$

δ_p为通带波纹或偏差，δ_s为阻带波纹或偏差。

7.6.3.2 带通滤波器（Mintzer and Liu，1979）

$$N \simeq \frac{C_\infty(\delta_p, \delta_s)}{\Delta F} + g(\delta_p, \delta_s)\Delta F + 1 \tag{7.15}$$

其中

$$C_\infty(\delta_p, \delta_s) = \log_{10}\delta_s[b_1(\log_{10}\delta_p)^2 + b_2\log_{10}\delta_p + b_3] + [b_4(\log_{10}\delta_p)^2 + b_5\log_{10}\delta_p + b_6]$$

$$g(\delta_p, \delta_s) = -14.6\log_{10}\left(\frac{\delta_p}{\delta_s}\right) - 16.9$$

$$b_1 = 0.012\,01;\qquad b_2 = 0.096\,64$$

$$b_3 = -0.513\,25;\qquad b_4 = 0.002\,03$$

$$b_5 = -0.5705;\qquad a_6 = -0.443\,14$$

ΔF是用抽样频率归一化的过渡带宽。

附录中给出了根据7.14式和7.15式计算N值的一个C语言程序。

7.6.4 最佳方法计算滤波器系数的程序总结

- **步骤1** 指定带沿频率（即通带和阻带频率）、通带波纹和阻带衰减（分贝或普通单位）以及抽样频率。
- **步骤2** 每个带沿频率通过除以抽样频率来归一化。
- **步骤3** 由通带波纹和阻带衰减（用普通单位表示）以及归一化的过渡带宽（参见下面的注释）根据7.14式和7.15式来估计滤波器长度N。通常，满足要求指标的N值要比从公式中得出的值稍大（2或3）。
- **步骤4** 根据通带波纹和阻带波纹之比（或阻带波纹与通带波纹之比），对每一个频带求出权值，用普通单位表示。对每一个频带用整数权值是非常方便的，例如通带波纹和阻带波纹分别是0.01和0.02的低通滤波器（通带波纹和阻带衰减分别为0.09 dB和30.5 dB），通带权值为3、阻带权值为1。带通滤波器在通带内的偏差（波纹）为0.001，而对每一个阻带的偏差为0.0105，那么通带的权值为21、阻带的每个权值为2。
- **步骤5** 将参数输入到最佳设计程序，从而得到系数N、带沿频率、每个频带的权值以及合适的网格密度（通常为16或32）。
- **步骤6** 检查由程序产生的通带波纹和阻带衰减。
- **步骤7** 若不满足规范，增加N值，重复步骤5和步骤6直至满足规范。然后求并且检查频率响应，确保满足规范。

应该注意的是，最佳方法在近似阶段只考虑了通带和阻带，而把过渡区域看成了不太关心的区域。为了确保成功，或者为了避免算法发散，当设计带通或多带滤波器时，最好是建立一个过渡区域，这个过渡区域等于最小过渡区域的带宽。如果采用不等过渡带宽，则应经常检查频率响应以确保其满足规范要求。在过渡频带可能出现局部最大值或局部最小值，从而给出意想不到的滤波器特性。

7.6.5　说明性的例子

下面的例子说明最佳程序的应用。

例 7.6　线性相位带通滤波器要求满足下列规范：

通带	900 ~ 1100 Hz
通带波纹	< 0.87 dB
阻带衰减	> 30 dB
抽样频率	15 kHz
过渡频率	450 Hz

利用最佳方法求合适的系数，画出滤波器的频谱。

解：

根据技术规范，滤波器有三个频带：下阻带（0 ~ 450 Hz）、通带（900 ~ 1100 Hz）、上阻带（1550 ~ 7500 Hz）。为了使用最佳设计程序，带沿频率必须归一化，以抽样频率的分数形式表示：

$$450 \to 450/15\,000 = 0.03$$

$$900 \to 900/15\,000 = 0.06$$

$$1100 \to 1100/15\,000 = 0.0733$$

$$1550 \to 1550/15\,000 = 0.1033$$

$$7500 \to 7500/15\,000 = 0.5$$

这样，三个归一化频带为(0 ~ 0.03)、(0.06 ~ 0.0733)、(0.1033 ~ 0.5)。

下面，我们必须为频带选择权值。权值取决于通带和阻带偏差。从给定的通带波纹和阻带衰减可求得用普通单位表示的偏差：

$$0.87\text{ dB 波纹：}20\log(1+\delta_p) \to \delta_p = 0.105\,35$$
$$30\text{ dB 衰减：}-20\log(\delta_s) \to \delta_s = 0.031\,623$$

δ_p 与 δ_s 的之比为 3.33 = 10/3：

$$\frac{\delta_p}{\delta_s} = \frac{10}{3} = \frac{\text{阻带权值}}{\text{通带权值}}$$

因此我们可以使用通带权值 3、阻带权值 10。通带权值 1 和阻带权值 3.33 是等效的，格网密度取 32。对于 N，应用实现的关系式程序，滤波器的长度求得为 40。我们采用 $N = 41$。

最佳程序的输入总结如下：

滤波器长度 N	41
滤波器类型 Jtype	1
权值 $W(\omega)$	10, 3, 10

Ngrid	32
边沿频率	0, 0.03, 0.06, 0.0733, 0.1033, 0.5

表 7.7 给出了打印的设计程序输出结果，图 7.13 给出了频谱。下面是一些注释：

- 通带偏差是阻带偏差的 3.33 倍，这是因为通带和阻带中的误差给定权值分别为 3 和 10。频带权值越大，得出波纹和偏差越小。
- 通带波纹和阻带衰减（用分贝表示）在技术规范内。
- 有 22 个极值频率，即在幅度响应中有$(N+3)/2$个最大值和最小值。注意，带沿频率$f=0$和$f=0.5$ Hz 也是极值频率，带沿频率总是极值频率。
- 冲激响应关于中间系数是对称的。对于线性相位响应来说对称是必要的。注意，对于类型 1 的滤波器，中间系数值是最大的。

表 7.7　最佳滤波器冲激响应系数（例 7.6）

```
H( 1) = -0.15346380E-01 = H(41)
H( 2) = -0.57805500E-04 = H(40)
H( 3) =  0.50234820E-02 = H(39)
H( 4) =  0.12667060E-01 = H(38)
H( 5) =  0.21082060E-01 = H(37)
H( 6) =  0.27764180E-01 = H(36)
H( 7) =  0.30053620E-01 = H(35)
H( 8) =  0.25869350E-01 = H(34)
H( 9) =  0.14445660E-01 = H(33)
H(10) = -0.31893230E-02 = H(32)
H(11) = -0.24161370E-01 = H(31)
H(12) = -0.44207120E-01 = H(30)
H(13) = -0.58574530E-01 = H(29)
H(14) = -0.63185570E-01 = H(28)
H(15) = -0.55754610E-01 = H(27)
H(16) = -0.36546910E-01 = H(26)
H(17) = -0.85400990E-02 = H(25)
H(18) =  0.23083860E-01 = H(24)
H(19) =  0.52013800E-01 = H(23)
H(20) =  0.72248070E-01 = H(22)
H(21) =  0.79516810E-01 = H(21)
```

	BAND 1	BAND 2	BAND 3
LOWER BAND EDGE	0.000000000	0.060000000	0.103300000
UPPER BAND EDGE	0.030000000	0.073300000	0.500000000
DESIRED VALUE	0.000000000	1.000000000	0.000000000
WEIGHTING	10.000000000	3.000000000	10.000000000
DEVIATION	0.028891690	0.096305620	0.028891690
RIPPLE IN DB	-30.784510000	0.798631800	-30.784510000

EXTREMA FREQUENCIES

0.0000000	0.0208333	0.0300000	0.0600000	0.1033000
0.1122285	0.1308297	0.1538951	0.1777045	0.2015139
0.2260674	0.2506209	0.2759184	0.3004719	0.3257694
0.3503229	0.3756204	0.4001739	0.4254714	0.4500249
0.4753224	0.5000000			

对于一个给定的设计来说，固定带沿频率，利用权值和N值，设计者可以相互比较通带波纹和阻带衰减并进行调整。

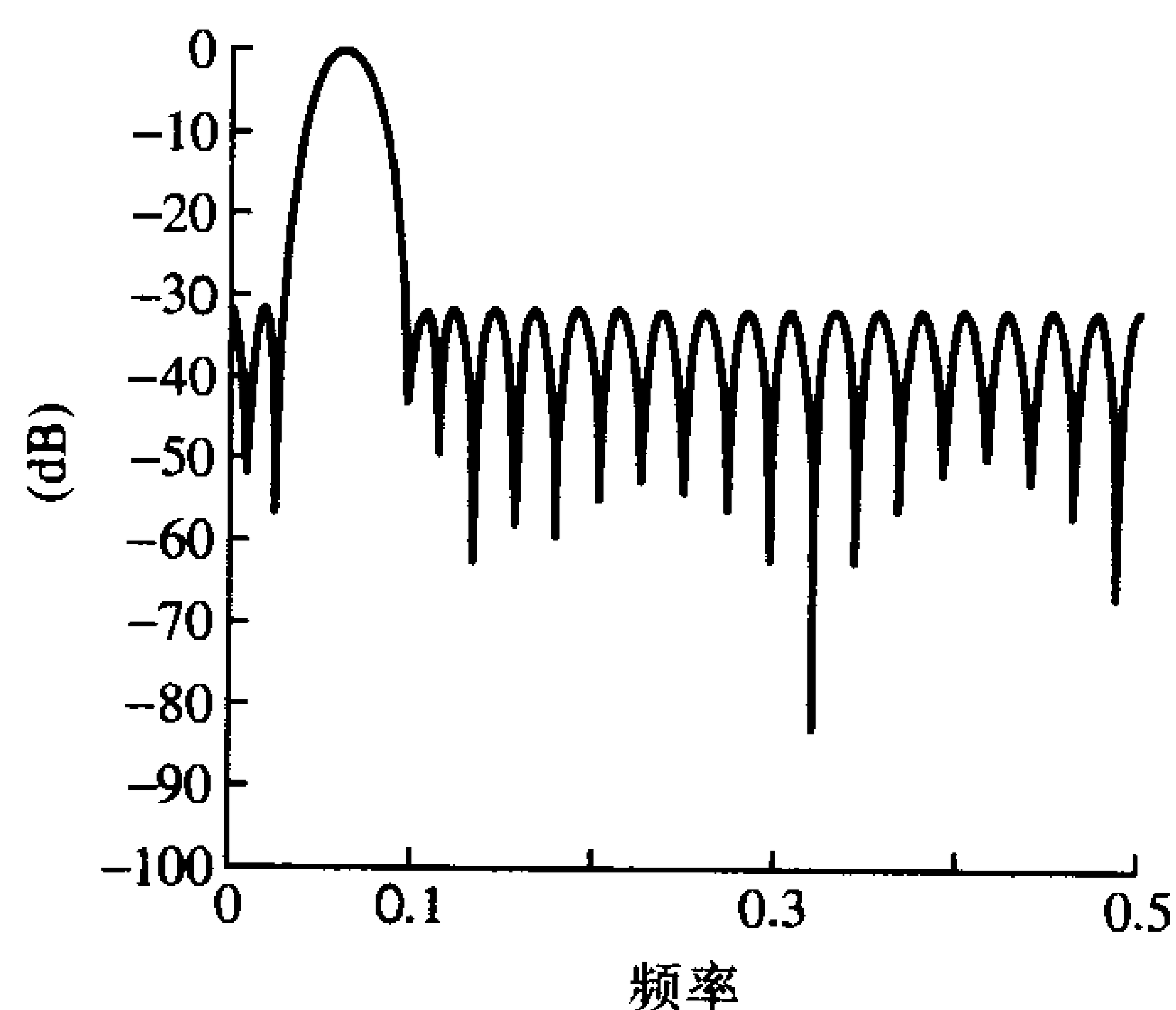


图7.13　滤波器的频率响应（归一化的频率尺度）

例7.7　数字FIR陷波滤波器应满足下面给出的规范要求：

陷波频率	1.875 kHz
在陷波频率处的衰减	> 60 dB
通带边沿频率	1.575 kHz和2.175 kHz
通带波纹	< 0.01 dB
抽样频率	7.5 kHz
系数数目	61

使用最佳方法求满足该规范的FIR滤波器的系数。

解：

滤波器有三个频带，这三个频带的归一化频率和偏差为

下通带	0 ~ 0.21
陷波频率	0.25
上通带	0.29 ~ 0.5
通带偏差	0.001 15（由 $20\log_{10}(1+\delta_p)$）
阻带偏差	0.001（由 $-20\log_{10}(\delta_s)$）

频带的权值为1、1.1519、1（由 δ_p/δ_s 得出），结果总结在表7.8和图7.14中。注意，对于一个陷波，阻带是一个有效的单一频率。

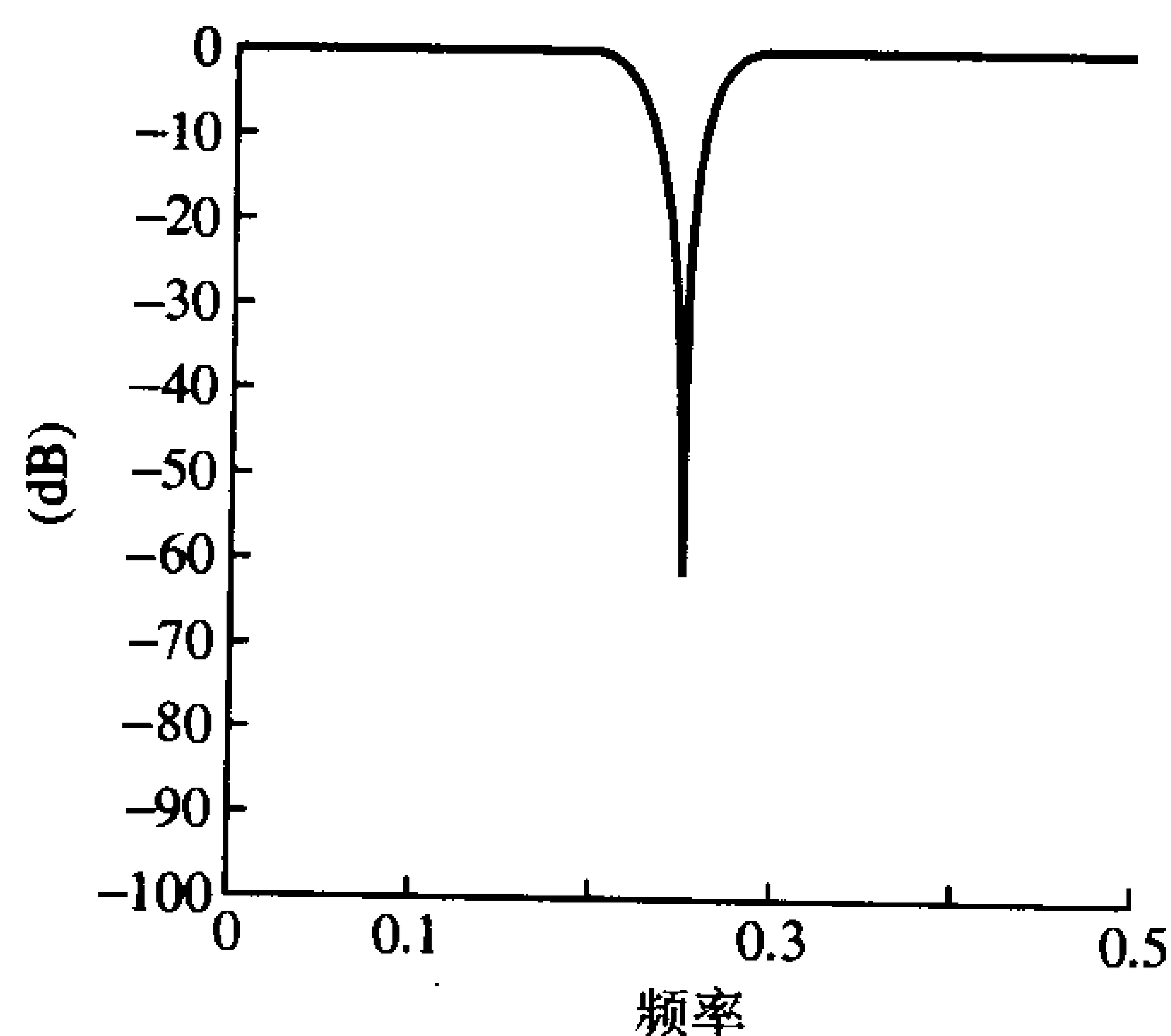


图7.14　滤波器响应（归一化的频率尺度）

因此，输入到设计程序的带沿频率是0、0.21、0.25、0.25、0.29和0.5。通过两次输入陷波频率，阻带被有效地化简成单一的频率，这正是我们期望得到的。

表7.8 最佳滤波器冲激响应系数（例7.7）

```
H( 1) =  0.12743640E-02 = H(61)
H( 2) =  0.26730640E-05 = H(60)
H( 3) = -0.23681110E-02 = H(59)
H( 4) = -0.17416350E-05 = H(58)
H( 5) =  0.43428480E-02 = H(57)
H( 6) =  0.53579250E-05 = H(56)
H( 7) = -0.71570240E-02 = H(55)
H( 8) = -0.49028620E-05 = H(54)
H( 9) =  0.10897540E-01 = H(53)
H(10) =  0.89629280E-05 = H(52)
H(11) = -0.15605960E-01 = H(51)
H(12) = -0.85508990E-05 = H(50)
H(13) =  0.21226410E-01 = H(49)
H(14) =  0.12250150E-04 = H(48)
H(15) = -0.27630130E-01 = H(47)
H(16) = -0.11091200E-04 = H(46)
H(17) =  0.34579770E-01 = H(45)
H(18) =  0.13800660E-04 = H(44)
H(19) = -0.41774130E-01 = H(43)
H(20) = -0.11560390E-04 = H(42)
H(21) =  0.48832790E-01 = H(41)
H(22) =  0.12787590E-04 = H(40)
H(23) = -0.55359840E-01 = H(39)
H(24) = -0.90065860E-05 = H(38)
H(25) =  0.60944450E-01 = H(37)
H(26) =  0.88997300E-05 = H(36)
H(27) = -0.65232190E-01 = H(35)
H(28) = -0.38167120E-05 = H(34)
H(29) =  0.67925720E-01 = H(33)
H(30) =  0.27041150E-05 = H(32)
H(31) =  0.93115220E+00 = H(31)
```

	BAND 1	BAND 2	BAND 3
LOWER BAND EDGE	0.000000000	0.250000000	0.290000000
UPPER BAND EDGE	0.210000000	0.250000000	0.500000000
DESIRED VALUE	1.000000000	0.000000000	1.000000000
WEIGHTING	1.000000000	1.151900000	1.000000000
DEVIATION	0.000978727	0.000849663	0.000978727
RIPPLE IN DB	0.008496785	-61.414990000	0.008496785

EXTREMA FREQUENCIES

0.0000000	0.0161290	0.0322580	0.0483871	0.0645161
0.0606451	0.0962701	0.1123991	0.1280241	0.1431450
0.1582660	0.1728829	0.1864918	0.1980845	0.2066530
0.2100000	0.2500000	0.2900000	0.2930243	0.3020971
0.3136902	0.3272994	0.3414126	0.3565342	0.3721596
0.3677850	0.4034105	0.4195400	0.4356695	0.4517990
0.4679265	0.4840580			

例7.8 设计者如何增加参数的交互性，以便必要时适当地调整它们，这是十分重要的。这个例子允许我们考察参数δ_p、δ_s和W的影响以及各种可能性。

对一个抑制心理噪声的线性相位 FIR 滤波器存在一定的要求（Hamer et al., 1985）。若滤波器是一个大的时间–临界（time-critical）DSP 系统，那么滤波器系数数目应尽可能小。滤波器特性应满足下面的规范：

通带波纹	< 0.026 dB
阻带	> 30 dB
通带边沿频率	10 Hz
阻带边沿	< 20 Hz
抽样频率	128 Hz

解：

归一化的带沿频率、通带偏差和阻带偏差是

通带边沿频率	0.078
阻带边沿频率	< 0.156 25
通带偏差	< 0.003
阻带偏差	> 0.0316

由于多数滤波器规范是可变的，很明显存在一个可能解的范围。那么问题就是找出其中一个最好的解。

在上面的 7.14 式中使用极限值，我们求出 $N > 25.6$（这代表 N 的最小可能值）。对于 25 ~ 37 中的每一个 N 值，使用下面的关系计算出满足规范的阻带频率 f_s：

$$f_s = f_p + \Delta f$$

其中 f_s 和 f_p 为阻带和通带的边沿频率，过渡带宽 Δf 由下面的公式给出（$\Delta f_{max} = 20 - 10\,\text{Hz} = 10\,\text{Hz}$）：

$$\Delta f = \frac{N-1}{2f(\delta_p, \delta_s)}\left[1 + \frac{4f(\delta_p, \delta_s)D_\infty(\delta_p, \delta_s) - 1}{(N-1)^2}\right]^{1/2}$$

图 7.15 给出了解空间（曲线上面），它是由阻带边沿频率 20 Hz 及 $N = 26$ 和 37 所围成的区域。选择 27 是一个好的解决方案。N 为奇数更能避免通过滤波器的非整数抽样延迟。采用下列参数：通带（0 ~ 0.078）、阻带（0.152 388 5 ~ 0.5，也就是 19 Hz ~ 64 Hz）。通带和阻带权值分别为 10.5 和 1。表 7.9 给出了得出的滤波器系数和参数。滤波器参数和频谱（未给出）说明它们满足规范。

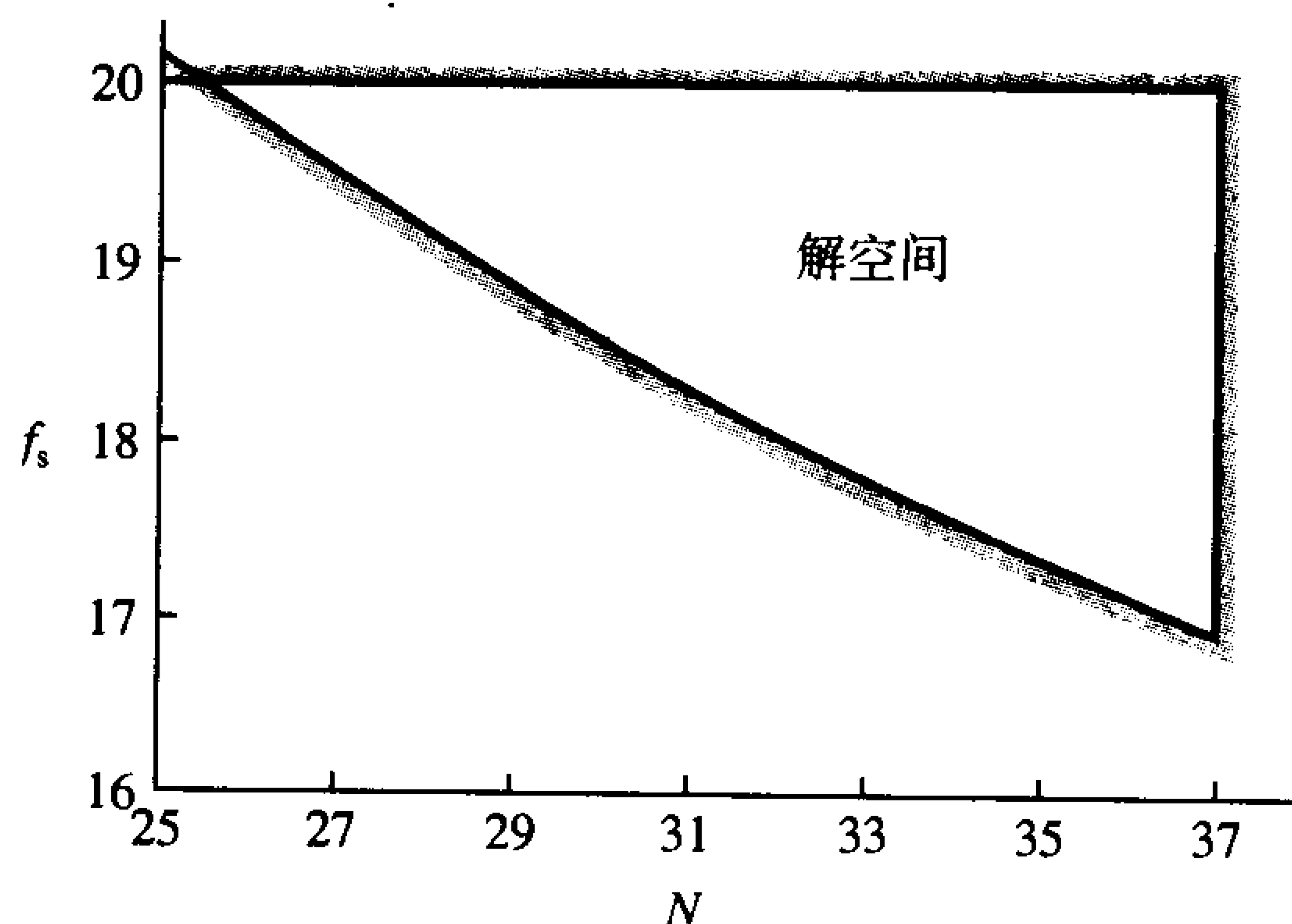


图 7.15　阻带频率与滤波器长度一起表示可能的解空间

表 7.9 最佳滤波器的冲激响应系数（例 7.8）

H(1) = −0.13614960E−01 = H(27)
H(2) = 0.34793330E−02 = H(26)
H(3) = 0.11140420E−01 = H(25)
H(4) = 0.16664540E−01 = H(24)
H(5) = 0.12807340E−01 = H(23)
H(6) = −0.33202110E−02 = H(22)
H(7) = −0.26167170E−01 = H(21)
H(8) = −0.42047790E−01 = H(20)
H(9) = −0.34767040E−01 = H(19)
H(10) = 0.55338630E−02 = H(18)
H(11) = 0.75072090E−01 = H(17)
H(12) = 0.15527810E+00 = H(16)
H(13) = 0.21933680E+00 = H(15)
H(14) = 0.24378330E+00 = H(14)

	BAND 1	BAND 2
LOWER BAND EDGE	0.000000000	0.152388500
UPPER BAND EDGE	0.078000000	0.500000000
DESIRED VALUE	1.000000000	0.000000000
WEIGHTING	10.500000000	1.000000000
DEVIATION	0.002604177	0.027343860
RIPPLE IN DB	0.022589730	−31.262770000

EXTREMA FREQUENCIES

0.0089286	0.0468750	0.0691964	0.0780000	0.1523885
0.1668974	0.1981473	0.2338614	0.2706916	0.3086379
0.3465842	0.3845305	0.4235928	0.4615391	0.5000000

7.7 频率抽样方法

频率抽样方法允许我们设计非递归FIR滤波器：标准频率选择性滤波器（低通、高通、带通滤波器）和具有任意频率响应的滤波器。频率抽样方法惟一吸引我们之处在于它也允许FIR滤波器的递归实现，因而也是一种计算有效的滤波器。利用某些限制，可以设计系数为简单整数的递归FIR滤波器，当只有基本算术运算时，在系统中可用标准的微处理器来实现，这一点就很诱人。

7.7.1 非递归频率抽样滤波器

图7.16(a)描述了一个滤波器的频率响应，假设我们想求该滤波器的FIR系数。我们可以按间隔 kF_s/N（$k = 0, 1, \ldots, N-1$）对频率响应取 N 个样本。由频率抽样值的DFT反变换可求得滤波器系数 $h(n)$：

$$h(n) = \frac{1}{N}\sum_{k=0}^{N-1} H(k)\,\mathrm{e}^{\mathrm{j}(2\pi/N)nk} \tag{7.16}$$

其中 $H(k)$（$k = 0, 1, \ldots, N-1$）是理想或目标频率响应的抽样值。

可以证明（参见例7.9），对于具有正的对称冲激响应的线性相位滤波器，滤波器系数 $h(n)$ 可写为（N 为偶数）：

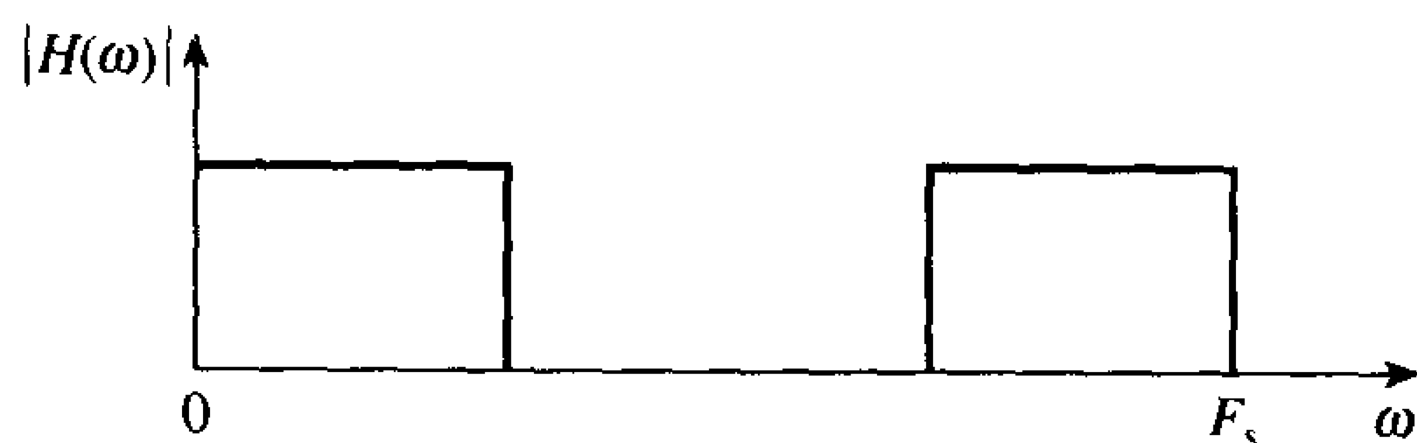
$$h(n) = \frac{1}{N}\left[\sum_{k=1}^{N/2-1} 2|H(k)|\cos[2\pi k(n-\alpha)/N] + H(0)\right] \tag{7.17}$$

其中 $\alpha=(N-1)/2$。N 为奇数时，求和的上限为$(N-1)/2$。得出的滤波器在抽样点处的频率响应与原始频率响应完全相同。而在抽样点之间的频率响应可能有很大的差别（参见图 7.16(c)）。为了求得期望的频率响应的一个好的近似，很明显，我们应取足够多的频率抽样值。

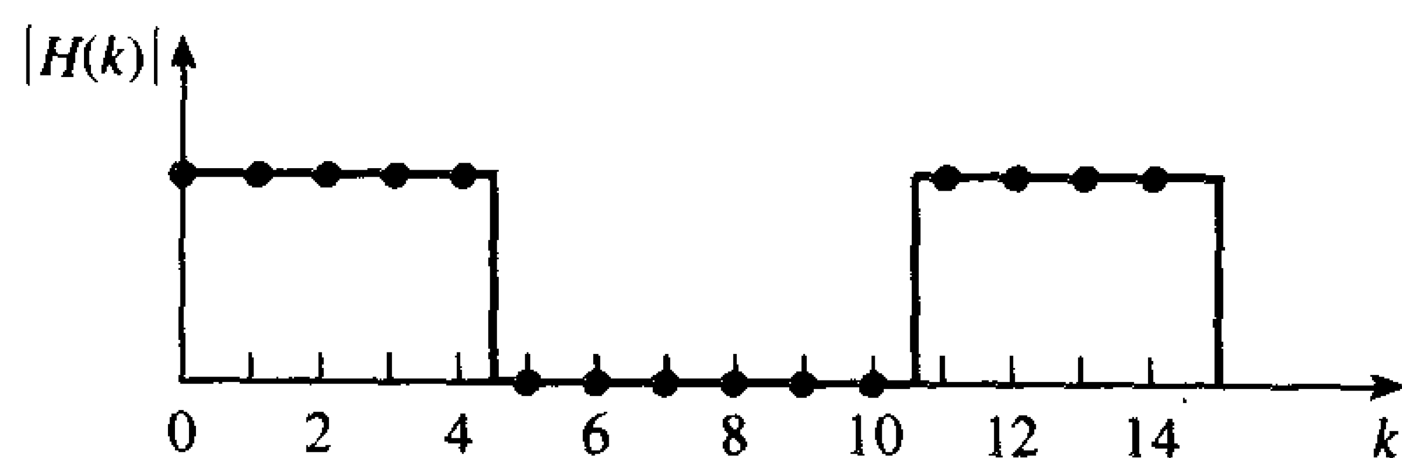
另一种频率抽样滤波器，称为类型 2，是以如下频率间隔取频率抽样值得出的：

$$f_k=(k+1/2)F_s/N,\quad k=0,1,\ldots,N-1 \tag{7.18}$$

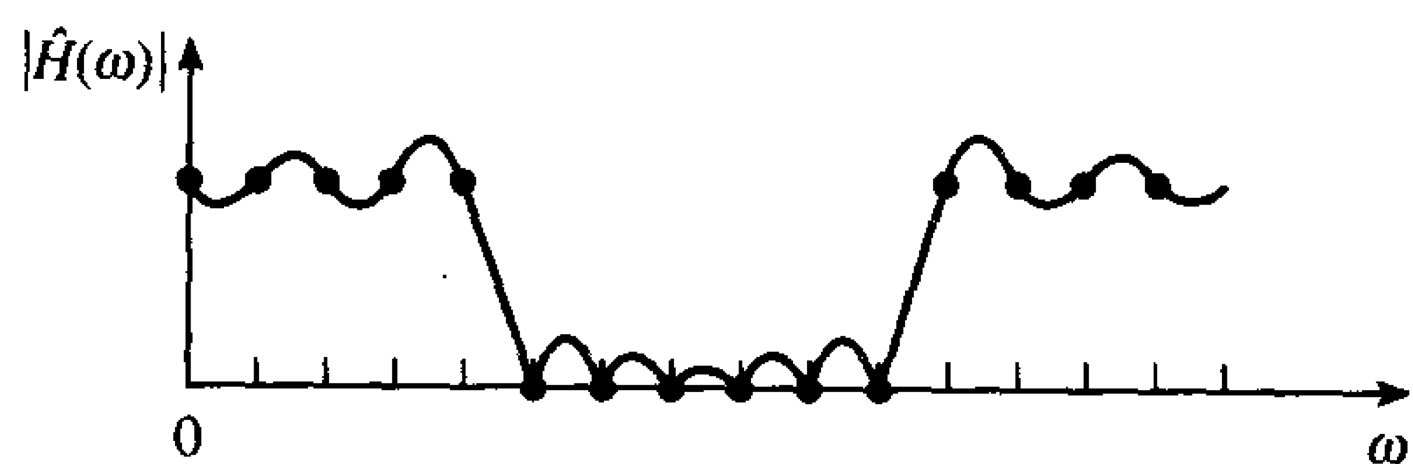
图7.17比较了两种类型的抽样方法的抽样栅格。对于一个给定的滤波器规范，两种方法导出的频率响应稍有点不同。设计者应根据自己的需要决定选择哪一种方法。



(a) 理想低通滤波器的频率响应



(b) 理想低通滤波器的抽样值



(c) 由(b)的频率抽样值推导出的低通滤波器的频率响应

图 7.16　频率抽样的概念

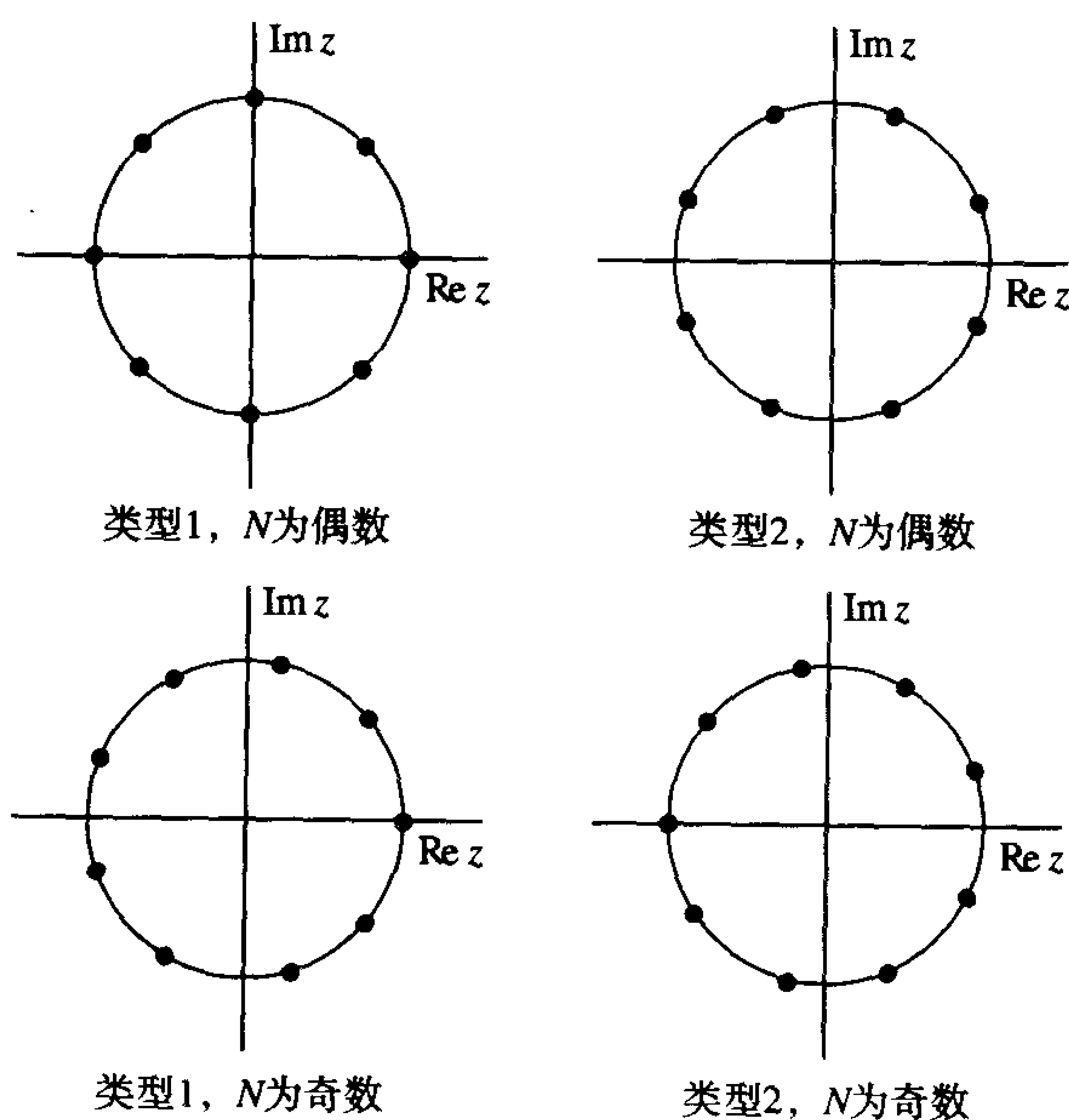


图 7.17　对于两种类型的频率抽样滤波器的四种可能的 z 平面抽样栅格

例 7.9

(1) 证明：正对称的线性相位 FIR 滤波器在 N 为偶数的情况下，其冲激响应系数可表示为

$$h(n) = \frac{1}{N}\left[\sum_{k=1}^{N/2-1} 2|H(k)|\cos[2\pi k(n-\alpha)/N] + H(0)\right]$$

其中 $\alpha=(N-1)/2$，$H(k)$是以间隔 kF_s/N 对滤波器的频率响应抽样的抽样值。

(2) 满足下列规范的低通 FIR 滤波器存在一定的要求：

通带	0 ~ 5 kHz
抽样频率	18 kHz
滤波器长度	9

使用频率抽样方法求滤波器系数。

解：

(1)

$$h(n) = \frac{1}{N}\sum_{k=0}^{N-1} H(k)\,e^{j(2\pi/N)nk} \tag{7.19}$$

$$= \frac{1}{N}\sum_{k=0}^{N-1} |H(k)|e^{-j2\pi\alpha k/N}e^{j2\pi kn/N}$$

$$= \frac{1}{N}\sum_{k=0}^{N-1} |H(k)|e^{j2\pi k(n-\alpha)/N}$$

$$= \frac{1}{N}\sum_{k=0}^{N-1} |H(k)|\cos[2\pi k(n-\alpha)/N] + j\sin[2\pi k(n-\alpha)/N]$$

$$= \frac{1}{N}\sum_{k=0}^{N-1} |H(k)|\cos[2\pi k(n-\alpha)/N] \tag{7.20}$$

由于冲激响应完全是实的，对于线性相位$h(n)$最重要的是其对称性，所以我们将其表达式写为

$$h(n) = \frac{1}{N}\left[\sum_{k=1}^{N/2-1} 2|H(k)|\cos[2\pi k(n-\alpha)/N] + H(0)\right] \tag{7.21}$$

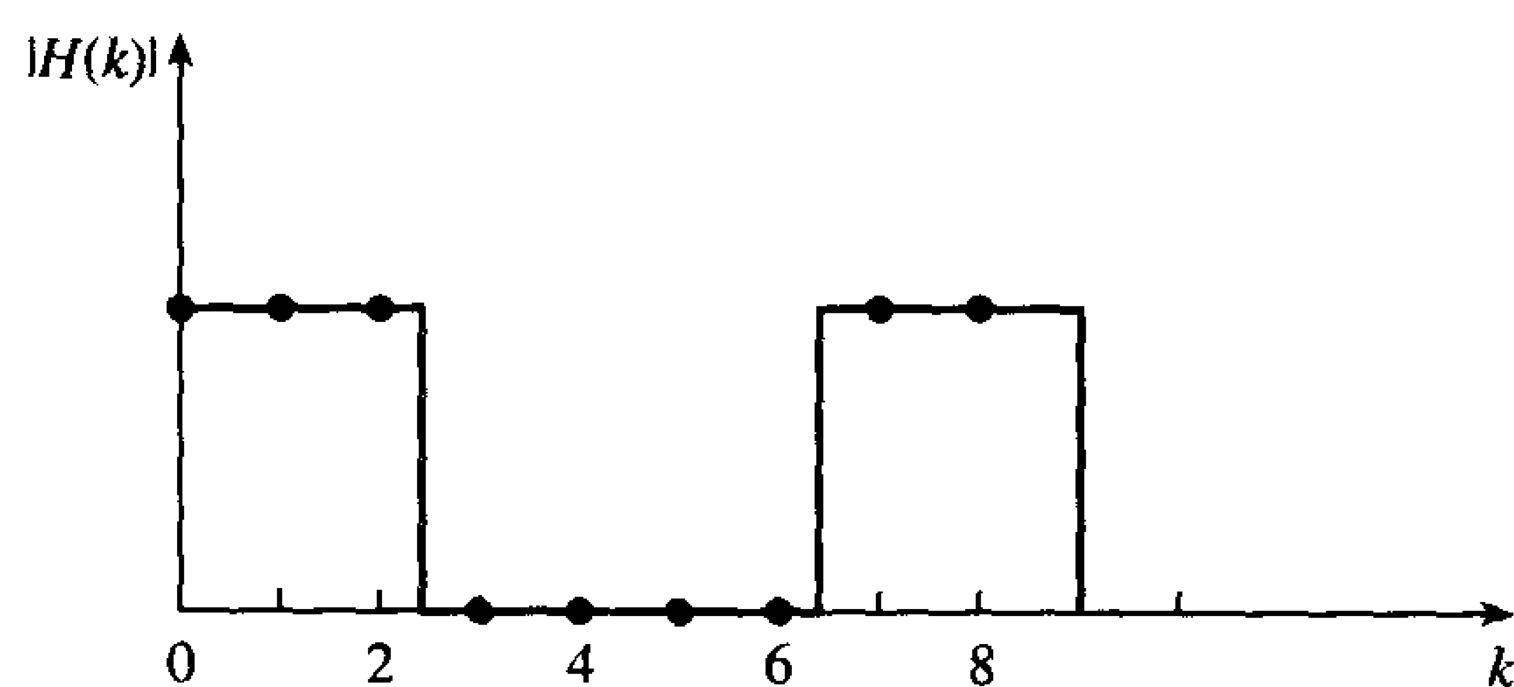
当 N 为奇数时，求和的上限为$(N-1)/2$。

(2) 7.18(a)图描述的为理想频率响应。以间隔 kF_s/N进行频率抽样，即间隔为 18/9 = 2 kHz。那么，频率抽样为

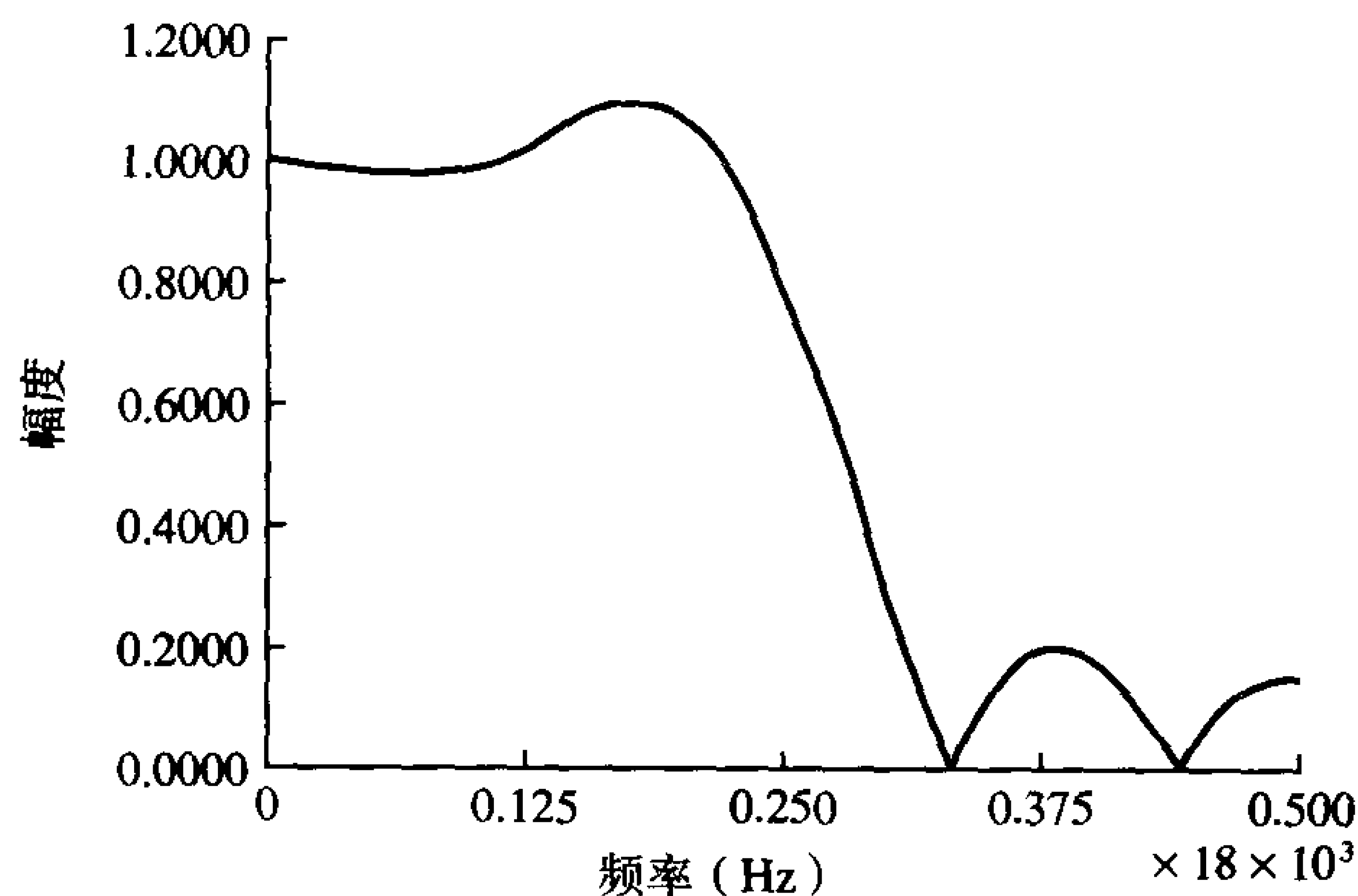
$$|H(k)| = 1 \quad k = 0, 1, 2$$
$$0 \quad k = 3, 4$$

利用 7.21 的上限$(N-1)/2$和频率抽样，我们求得冲激响应系数（参见表 7.10）。

在指导手册的CD中（参见前言）包含了一个在给定滤波器抽样值的情况下计算FIR 滤波器系数的程序。图7.18给出了滤波器的频率响应，从该图我们看出滤波器的幅度-频率响应较差，这是由从通带（$|H(k)|=1$）到阻带（$|H(k)|=0$）的突然变化引起的。



(a) 标明了抽样点的理想频率响应



(b) 频率抽样滤波器的频率响应

图 7.18　滤波器的频率响应

表 7.10　例 7.9 非递归 FIR 滤波器的系数

h[0] =	7.2522627e−02	= h[8]
h[1] =	−1.1111111e−01	= h[7]
h[2] =	−5.9120987e−02	= h[6]
h[3] =	3.1993169e−01	= h[5]
h[4] =	5.5555556e−01	= h[4]

7.7.1.1　最佳幅度响应

以上问题与矩形窗的问题类似。我们回想一下，在窗口方法中，为了改善幅度响应，我们可以折中考虑过渡带宽。为了改善频率抽样滤波器的幅度响应，以宽的过渡带宽为代价，我们可以在过渡频带引入频率抽样。图 7.19 描述了具有三个过渡带频率抽样值的低通滤波器的典型规范。对于低通滤波器，在过渡带宽内的每一个过渡带频率抽样值（Rabiner et al., 1970），阻带衰减大约增加20分贝：

阻带衰减的近似值 $(25+20M)$ dB

过渡带宽的近似值 $(M+1)F_s/N$

其中 M 为过渡带频率抽样值的个数，N 为滤波器长度。

给出最佳阻带衰减的过渡带频率抽样值由最佳过程来确定（Rabiner et al., 1970）。一个有用的最佳化目标是找出使峰值阻带波纹最小（即使阻带衰减最小）的过渡带频率抽样值 $T_1, T_2,...,T_M$。用数学公式表示为

$$\underset{\{T_1,T_2,...,T_M\}}{\text{最小化}} \begin{bmatrix} \max\left|W[H_D(\omega)-H(\omega)]\right| \\ \{\text{在阻带内的 } \omega\} \end{bmatrix} \tag{7.22}$$

其中 $H_D(\omega)$ 和 $H(\omega)$ 分别是滤波器理想频率响应和实际频率响应，W 为加权因子。

Rabiner et al.(1970)已经提供了一个最佳（7.22 式意义下的最佳）过渡带频率抽样值表，这个表已经被广泛使用。表 7.11 给出了 $N = 15$ 时的最佳过渡带频率抽样值。表中的带宽是指滤波器通带内的频率抽样点数。

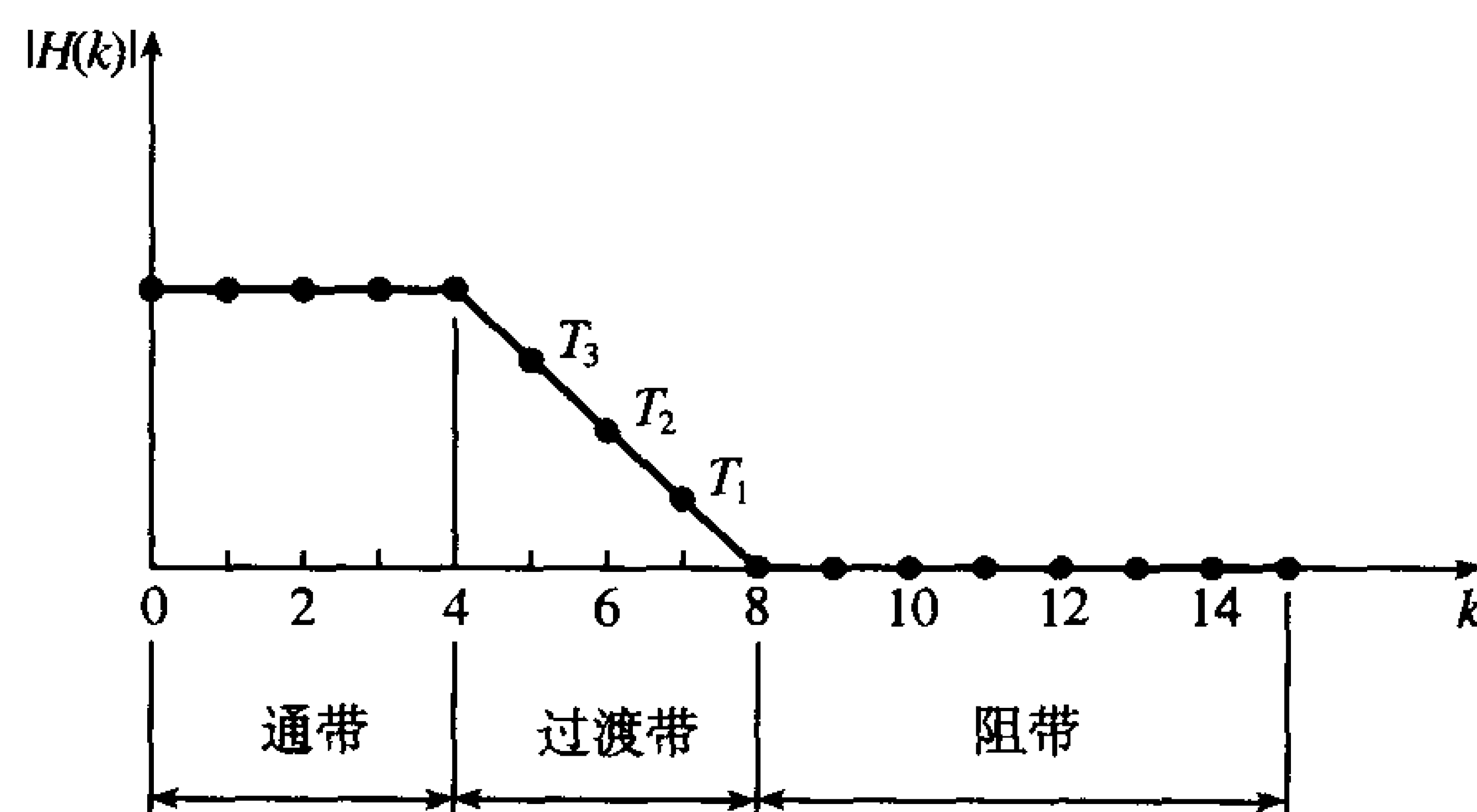


图 7.19　包含三个过渡带抽样的低通滤波器频率抽样值。注意：由于幅度响应的对称性，这里仅给出滤波器响应的一半

表 7.11　$N = 15$ 时类型 1 的低通频率抽样滤波器的最佳过渡带频率抽样值表（取自 Rabiner et al., 1970）

BW	阻带衰减（dB）	T_1	T_2	T_3
一个过渡带频率抽样值，$N = 15$				
1	42.309 322 83	0.433 782 96		
2	41.262 992 86	0.417 938 23		
3	41.253 337 86	0.410 473 63		
4	41.949 077 13	0.404 058 84		
5	44.371 245 38	0.392 681 89		
6	56.014 165 88	0.357 665 25		
两个过渡带频率抽样值，$N = 15$				
1	70.605 405 85	0.095 001 22	0.589 954 18	
2	69.261 681 56	0.103 198 24	0.593 571 18	
3	69.919 734 95	0.100 836 18	0.589 432 70	
4	75.511 722 56	0.084 074 93	0.557 153 12	
5	103.460 783 00	0.051 802 06	0.499 174 24	
三个过渡带频率抽样值，$N = 15$				
1	94.611 661 91	0.014 550 78	0.184 578 82	0.668 976 13
2	104.998 130 80	0.010 009 77	0.173 607 13	0.659 515 26
3	114.907 193 18	0.008 734 13	0.163 973 10	0.647 112 64
4	157.292 575 84	0.003 787 99	0.123 939 63	0.601 811 54

BW 指通带内频率抽样的个数。

大多数情况下，过渡带频率抽样值通常在下面的范围内：

对于一个过渡带频率抽样值，

$$0.250 < T_1 < 0.450$$

对两个过渡带频率抽样值，

$$0.040 < T_1 < 0.150$$

$$0.450 < T_2 < 0.650$$

对于三个转移频率抽样点，

$$0.003 < T_1 < 0.035$$

$$0.100 < T_2 < 0.300$$

$$0.550 < T_3 < 0.750$$

对于宽带滤波器来说，值越小将越导致阻带衰减越大。

例 7.10

(1) 下列频率抽样值刻画了线性相位 15 点 FIR 滤波器：

$$|H(k)| = 1 \quad k = 0, 1, 2, 3$$
$$0 \quad k = 4, 5, 6, 7$$

假设抽样频率为 2 kHz，求它的频率响应。

(2) 比较滤波器的频率响应，如果(a)用一个过渡带频率抽样值，(b)用两个过渡带频率抽样值，(c)用三个过渡带频率抽样值。

解：

(1) 用频率抽样值作为输入加到设计程序 fresamp.c（参见附录）中，表 7.12 的第 2 列给出了滤波器的系数，图 7.20(a)给出了相应的频率响应。

(2) 对于(a)的情况，由表 7.11 看出，过渡带频率抽样值是 0.4041。因此，滤波器的频率抽样值为

$$|H(k)| = 1 \quad k = 0, 1, 2, 3$$
$$0.404\,06 \quad k = 4$$
$$0 \quad k = 5, 6, 7$$

用这些频率抽样值作为输入加到设计程序中，计算出滤波器系数值，总结在见表 7.12 中。相应的频率响应由图 7.20(b)给出。

表 7.12　对于不同的过渡带频率抽样值的非递归滤波器系数值

	没有过渡带抽样值	一个过渡带抽样值	两个过渡带抽样值	三个过渡带抽样值
h[0] =	−4.9815884e−02	−1.3766696e−02	−5.7195305e−03	−4.2282741e−03
h[1] =	4.1202267e−02	−2.3832554e−03	−7.6781827e−03	−7.6031627e−03
h[2] =	6.6666666e−02	3.9729333e−02	2.3920000e−02	1.8793332e−02
h[3] =	−3.6487877e−02	1.2729081e−02	2.5763613e−02	2.8145113e−02
h[4] =	−1.0786893e−01	−9.1220745e−02	−7.3701817e−02	−6.6396840e−02
h[5] =	3.4078020e−02	−1.8619356e−02	−4.4185450e−02	−5.2511978e−02
h[6] =	3.1889241e−01	3.1326097e−01	3.0552137e−01	3.0183514e−01
h[7] =	4.6666667e−01	5.2054133e−01	5.5216000e−01	5.6393334e−01

由于对称性，这里仅列出系数的前一半。

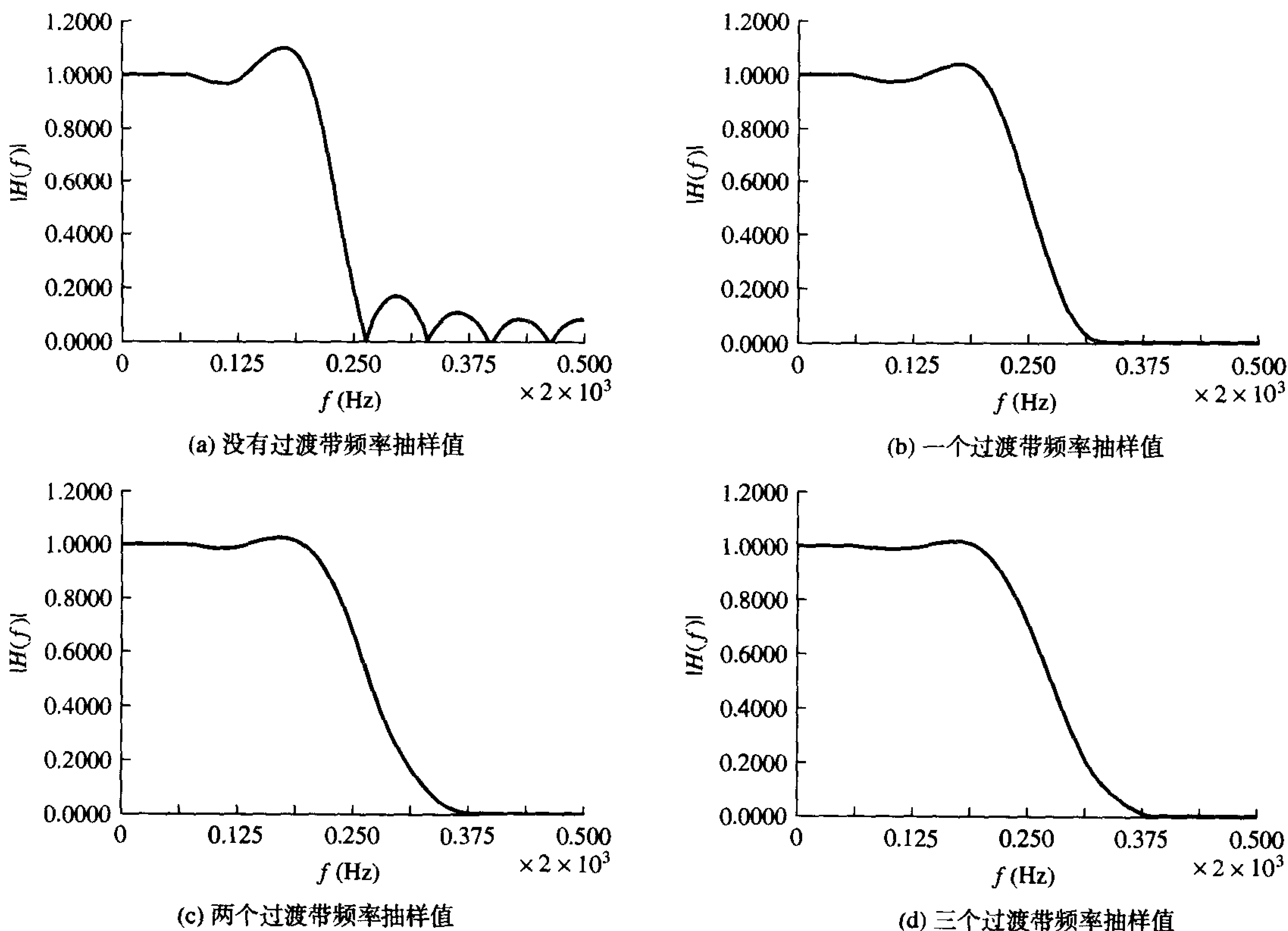
对于(b)和(c)，频率抽样值分别定义为

$$|H(k)| = 1 \quad k = 0, 1, 2, 3$$
$$0.5571 \quad k = 4$$
$$0.0841 \quad k = 5$$
$$0 \quad k = 6, 7$$

$$|H(k)| = 1 \quad k = 0, 1, 2, 3$$
$$0.6018 \quad k = 4$$
$$0.1239 \quad k = 5$$
$$0.0038 \quad k = 6$$
$$0 \quad k = 7$$

这些情况下的系数总结在表7.12的第4列和第5列中。图7.20(c)和图7.20(d)给出其相应的频率响应。从图中可以看出，随着过渡带频率抽样点数的增加，幅度响应得到改善（就通带和阻带波纹来说），但是以增加过渡带宽或频率响应的跌落作为代价。

可以用来改善幅度响应的另一种方法是通过以紧凑的间隔抽样求出许多频率抽样值，用7.21式计算冲激响应，然后应用前面讨论的一种窗口函数将滤波器长度降低到期望的长度。



(a) 没有过渡带频率抽样值

(b) 一个过渡带频率抽样值

(c) 两个过渡带频率抽样值

(d) 三个过渡带频率抽样值

图7.20 频率抽样滤波器的频率响应

7.7.1.2 频率抽样滤波器的自动设计

如前面叙述的那样，过渡带频率抽样最佳值表已在文献（Rabiner et al., 1970）中提供，而且广泛应用到频率抽样滤波器设计上。如果设计者要设计一个表中没有列出的滤波器，那么可以通过线性内插求出过渡带频率抽样近似值，但这并不总是行得通，尤其是设计包含了大量过渡带抽样点的时候。而且表中的信息不是设计者熟悉的形式，例如没有给出带沿频率和通带波纹。通用计算机程序目前已达到自动设计递归和非递归频率抽样滤波器（Ifeachor and Harris, 1993; Harris and Ifeachor, 1998）设计的许多方面。从本质上讲，在程序中过渡带抽样值是通过混合遗传算法（GA）达到最优的，通过这种方法对于指定的一组滤波器规范给出了阻带中的最大衰减。依靠文献中列出的表验证了这一方法，并且发现在每种情况下，这种方法的结果相同或有所改善，而且这一方法还允许设计表中没有列出的滤波器。

例7.11 求最佳过渡带频率抽样值和对应的满足下面规范的低通滤波器的滤波器系数：

通带边沿频率	0.143（规一化）
阻带边沿频率	0.245（规一化）
滤波器系数数目	49

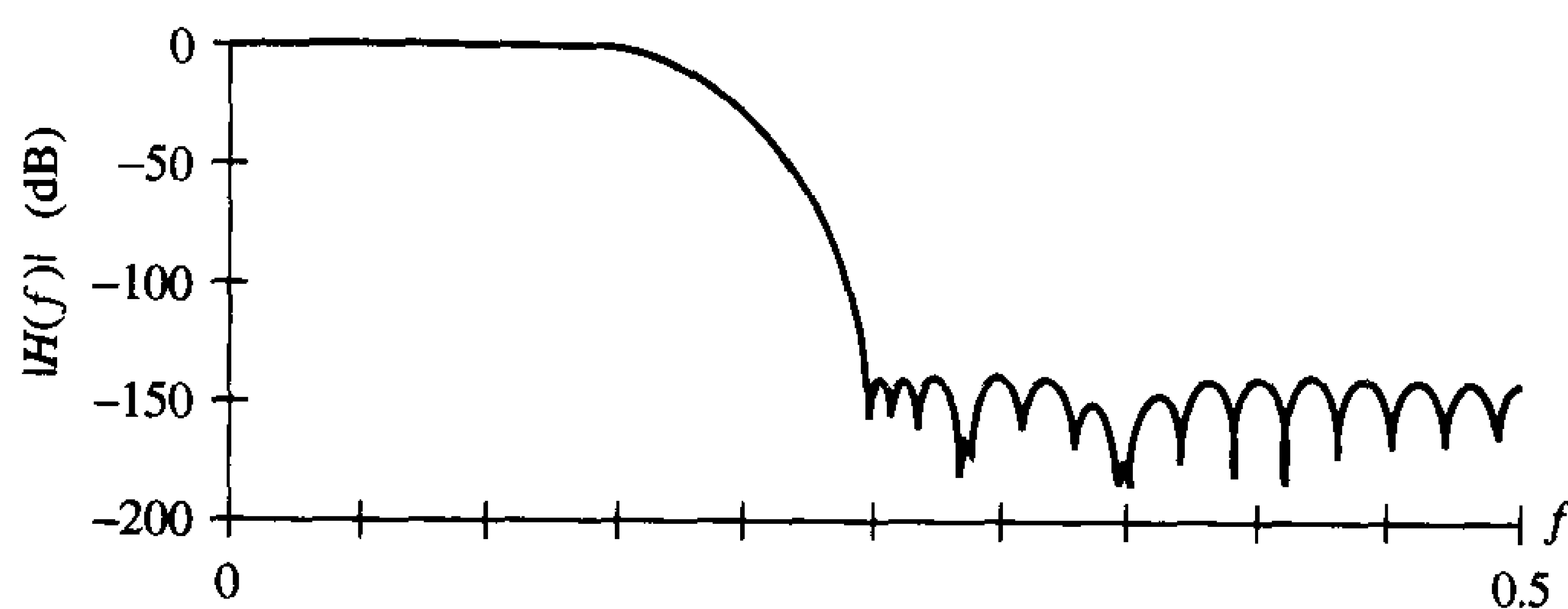
解：

根据规范，频率抽样点数 $N=49$，对应于通带和阻带边沿频率的抽样点数分别为 6 和 12，过渡带抽样点数 $M=5$；则理想幅度－频率响应的频率抽样值为

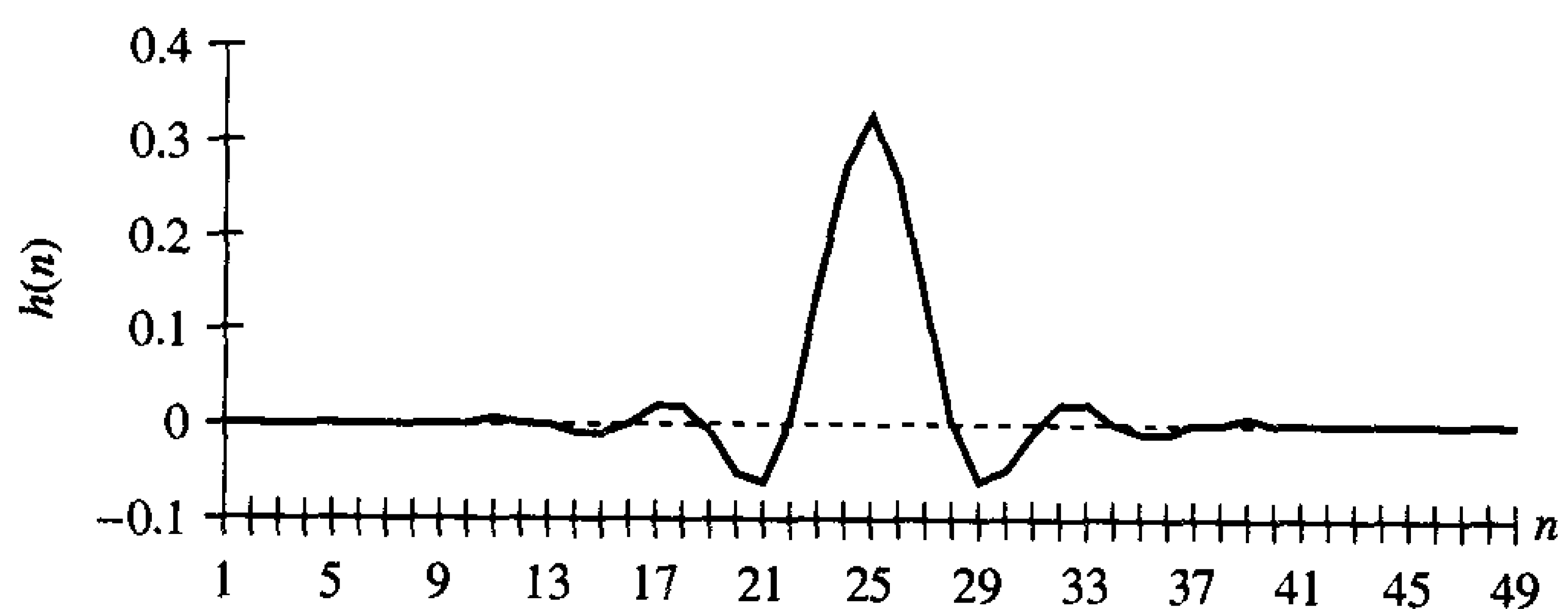
$$
\begin{aligned}
|H(k)| &= 1, & k&=0,1,\ldots,6\\
& T_{k-6}, & k&=7,\ldots,11\\
& 0 & k&=12,\ldots,24
\end{aligned}
$$

T_1 和 T_5 值没有指定，我们可以通过最佳方法的混合 GA 程序求出这些值。最佳化过程的结果总结在图 7.21 中。

虽然混合GA方法得到的结果比文献中的结果有所改善，但它的最大优点还是能快速产生表中没有列出的滤波器的系数，这要比由内插方法得到的结果更有意义。使用该方法也能设计具有多个过渡带抽样值的滤波器。



(a) 内插频率响应



(b) 滤波器系数。通带波纹：0.046 dB；阻带衰减：139.64 dB；通带宽度：0.15；5个过渡带抽样值；滤波器系数数目：49；过渡带抽样值为：0.855 456, 0.485 507, 0.148 961, 0.019 693, 0.000 644

图 7.21　最佳化过程的结果

7.7.2　递归频率抽样滤波器

如果频率抽样值中有许多零值，那么频率抽样滤波器的递归形式要比非递归形式更具有计算上的优势。可以证明（参见例 7.12），FIR 滤波器的传递函数 $H(z)$ 的递归形式表示为

$$
H(z)=\frac{1-z^{-N}}{N}\sum_{k=0}^{N-1}\frac{H(k)}{1-\mathrm{e}^{\mathrm{j}2\pi k/N}z^{-1}}=H_1(z)H_2(z) \tag{7.23}
$$

其中

$$
H_1(z)=\frac{1-z^{-N}}{N}
$$

$$
H_2(z)=\sum_{k=0}^{N-1}\frac{H(k)}{1-\mathrm{e}^{\mathrm{j}2\pi k/N}z^{-1}}
$$

因此我们看到，用递归形式表示的$H(z)$可以看成两个滤波器的级联：一个是梳状滤波器$H_1(z)$，它的N个零点均匀分布在单位圆上；另一个$H_2(z)$是N个单个全极点滤波器之和。梳状滤波器的零点和单极滤波器的极点在$z_k = e^{j2\pi k/N}$的单位圆上是重合的。因此，零点与极点相消，使得$H(z)$成了一个没有极点的FIR滤波器。

实际上，有限字长效应使得$H_2(z)$的极点不能精确地定位在单位圆上，所以它们并不能与零点对消，使得$H(z)$是一个IIR，且存在潜在的不稳定。稳定性问题可以通过比单位圆稍小的半径r对$H(z)$进行抽样避免，这种情况下传递函数$H(z)$为

$$H(z) = \frac{1 - r^N z^{-N}}{N} \sum_{k=0}^{N-1} \frac{H(k)}{1 - r e^{j2\pi k/N} z^{-1}} \tag{7.24}$$

一般来说，频率抽样值$H(k)$是复的。因此，直接实现7.23式和7.24式要求复杂算术运算。为避免这种复杂性，我们充分利用具有实时冲激响应$h(n)$的FIR滤波器的频率特性固有的对称性。对于标准频率选择线性相位滤波器（正对称冲激响应），7.24式可表示为

$$H(z) = \frac{1 - r^N z^{-N}}{N} \times \left[\sum_{k=1}^{M} \frac{|H(k)|\{2\cos(2\pi k\alpha/N) - 2r\cos[2\pi k(1+\alpha)/N]z^{-1}\}}{1 - 2r\cos(2\pi k/N)z^{-1} + r^2 z^{-2}} + \frac{H(0)}{1 - z^{-1}} \right] \tag{7.25}$$

其中$\alpha = (N-1)/2$。当N为奇数时，$M = (N-1)/2$，而当N为偶数时，$M = N/2-1$。图7.22描述了7.25式的实现框图。

7.7.3 简单系数的频率抽样滤波器

FIR滤波器的递归实现大大降低了数字滤波器中的算术运算量。另外，如果滤波器的系数是简单整数（或二次方）的，将会大大地改善计算效率。这使得它们在带有基本算术运算的处理器的应用中很有吸引力，例如在一般的微处理器上的应用。Lynn(1975)已开发出一种带有小整数系数的频率抽样滤波器。

然而，如果对传递函数（7.25式）的极点的位置给一定的约束，使用整数系数是有可能的。等价地，具有整数系数的滤波器通带可以以约束频率为中心。注意，由于系数为整数，我们可以将极点放在单位圆上，并且得到完美的对消。这些滤波器是频率抽样滤波器的特殊情况。

例 7.12

(1) FIR滤波器的传递函数定义为

$$H(z) = \sum_{n=0}^{N-1} h(n) z^{-n} \tag{7.26a}$$

由上面公式开始，证明：对于具有正对称冲激响应的线性相位FIR滤波器，$H(z)$可表示成下面的递归形式：

$$H(z) = \frac{1 - r^N z^{-N}}{N} \times \left[\sum_{k=1}^{M} \frac{|H(k)|\{2\cos(2\pi k\alpha/N) - 2r\cos[2\pi k(1+\alpha)/N]z^{-1}\}}{1 - 2r\cos(2\pi k/N)z^{-1} + r^2 z^{-2}} + \frac{H(0)}{1 - z^{-1}} \right]$$

其中 $\alpha=(N-1)/2$，$H(k)$是滤波器频率响应的抽样值，抽样间隔为 kF_s/N。

(2) 满足下列规范的低通滤波器存在一定的要求：

通带	0～4 kHz
抽样频率	18 kHz
滤波器长度	9

用频率抽样方法求递归形式的滤波器传递函数。假设半径 $r=1$，画出滤波器的实现框图，并且与直接形式的 FIR 比较计算的复杂性。

解：

(1) 根据频率抽样，滤波器的冲激响应定义如下：

$$h(n)=\frac{1}{N}\sum_{k=0}^{N-1}H(k)r^n\mathrm{e}^{\mathrm{j}2\pi nk/N}\quad k=0,1,\ldots,N-1,r\leqslant 1 \tag{7.26b}$$

将 7.26b 式代入 7.26a 式，传递函数 $H(z)$变为

$$H(z)=\sum_{n=0}^{N-1}h(n)z^{-n}=\sum_{n=0}^{N-1}\left[\frac{1}{N}\sum_{k=0}^{N-1}H(k)r^n\mathrm{e}^{\mathrm{j}2\pi nk/N}\right]z^{-n}$$

交换两个求和的顺序，我们得到

$$H(z)=\frac{1}{N}\sum_{k=0}^{N-1}H(k)\left\{\sum_{n=0}^{N-1}[r\mathrm{e}^{\mathrm{j}(2\pi k/N)}z^{-1}]^n\right\} \tag{7.27}$$

现在，可以用有限的几何级数表示为

$$S_N=\sum_{n=0}^{N-1}\delta^n=\frac{1-\delta^N}{1-\delta}\quad \delta\neq 1$$

在这种情况下，$\delta=r\mathrm{e}^{\mathrm{j}2\pi k/N}z^{-1}$，我们可以写成

$$\sum_{n=0}^{N-1}[r\mathrm{e}^{\mathrm{j}(2\pi k/N)}z^{-1}]^n=\frac{1-(r\mathrm{e}^{\mathrm{j}2\pi k/N}z^{-1})^N}{1-r\mathrm{e}^{\mathrm{j}2\pi k/N}z^{-1}}=\frac{1-r^N\mathrm{e}^{\mathrm{j}2\pi k}z^{-N}}{1-r\mathrm{e}^{\mathrm{j}2\pi k/N}z^{-1}}$$

$$=\frac{1-r^Nz^{-N}}{1-r\mathrm{e}^{\mathrm{j}2\pi k/N}z^{-1}}$$

由于 $\mathrm{e}^{\mathrm{j}2\pi k}=\cos(2\pi k)=1$，$k=0,1,\ldots$，因此我们可以将 7.27 式写成

$$H(z)=\frac{1-r^Nz^{-N}}{N}\sum_{k=0}^{N-1}\frac{H(k)}{1-r\mathrm{e}^{\mathrm{j}2\pi k/N}z^{-1}}=H_1(z)H_2(z) \tag{7.28}$$

其中

$$H_1(z)=\frac{1-r^Nz^{-N}}{N}$$

$$H_2(z)=\sum_{k=0}^{N-1}\frac{H(k)}{1-r\mathrm{e}^{\mathrm{j}2\pi k/N}z^{-1}}$$

将 $H_2(z)$展开，我们有如下形式：

$$H_2(z)=\frac{H(0)}{1-rz^{-1}}+\frac{H(1)}{1-r\mathrm{e}^{\mathrm{j}2\pi/N}z^{-1}}+\frac{H(2)}{1-r\mathrm{e}^{\mathrm{j}2\pi 2/N}z^{-1}}$$
$$+\ldots+\frac{H(N-2)}{1-r\mathrm{e}^{\mathrm{j}2\pi(N-2)/N}z^{-1}}+\frac{H(N-1)}{1-r\mathrm{e}^{\mathrm{j}2\pi(N-1)/N}z^{-1}}$$

对于具有实系数的滤波器，满足下列对称条件：

$$H(k)=H^*(N-k),\ \mathrm{e}^{\mathrm{j}2\pi(N-k)/N}=\mathrm{e}^{-\mathrm{j}2\pi k/N}$$

因此，我们可以将$H_2(z)$重写为

$$H_2(z)=\frac{H(0)}{1-rz^{-1}}+\frac{H(1)}{1-r\mathrm{e}^{\mathrm{j}2\pi/N}z^{-1}}+\frac{H(2)}{1-r\mathrm{e}^{\mathrm{j}2\pi 2/N}z^{-1}}$$
$$+\ldots+\frac{H^*(2)}{1-r\mathrm{e}^{-\mathrm{j}2\pi 2/N}z^{-1}}+\frac{H^*(1)}{1-r\mathrm{e}^{-\mathrm{j}2\pi/N}z^{-1}}$$

因此，极点共轭成对出现（不包括N为奇数时$k=0$处的极点，以及N为偶数时$k=0$、$k=N/2$处的极点）。对于长度为偶数的线性相位滤波器，$H(N/2)=0$。将第k个单极点部分和它的共轭部分合并，我们有

$$\frac{H(k)}{1-r\mathrm{e}^{\mathrm{j}2\pi k/N}z^{-1}}+\frac{H^*(k)}{1-r\mathrm{e}^{-\mathrm{j}2\pi k/N}z^{-1}}$$
$$=\frac{H(k)(1-r\mathrm{e}^{-\mathrm{j}2\pi k/N}z^{-1})+H^*(k)(1-r\mathrm{e}^{\mathrm{j}2\pi k/N}z^{-1})}{(1-r\mathrm{e}^{\mathrm{j}2\pi k/N}z^{-1})(1-r\mathrm{e}^{-\mathrm{j}2\pi k/N}z^{-1})} \tag{7.29}$$

简化分母，得

$$(1-r\mathrm{e}^{\mathrm{j}2\pi k/N}z^{-1})(1-r\mathrm{e}^{-\mathrm{j}2\pi k/N}z^{-1})=1-2r\cos(2\pi k/N)z^{-1}+r^2z^{-2} \tag{7.30}$$

对于具有正对称冲激响应的线性相位滤波器，$H(k)$为

$$H(k)=|H(k)|\,\mathrm{e}^{-\mathrm{j}2\pi k\alpha/N}$$

其中$\alpha=(N-1)/2$。因此，分子可简化为

$$|H(k)|\,\mathrm{e}^{-\mathrm{j}2\pi k\alpha/N}(1-r\mathrm{e}^{-\mathrm{j}2\pi k/N}z^{-1})+|H(k)|\mathrm{e}^{\mathrm{j}2\pi k\alpha/N}(1-r\mathrm{e}^{\mathrm{j}2\pi k/N}z^{-1})$$
$$=|H(k)|[\mathrm{e}^{-\mathrm{j}2\pi k\alpha/N}(1-r\mathrm{e}^{-\mathrm{j}2\pi k/N}z^{-1})+\mathrm{e}^{\mathrm{j}2\pi k\alpha/N}(1-r\mathrm{e}^{\mathrm{j}2\pi k/N}z^{-1})]$$
$$=|H(k)|(\mathrm{e}^{-\mathrm{j}2\pi k\alpha/N}-r\mathrm{e}^{-\mathrm{j}2\pi k\alpha/N}\mathrm{e}^{-\mathrm{j}2\pi k/N}z^{-1}+\mathrm{e}^{\mathrm{j}2\pi k\alpha/N}-r\mathrm{e}^{\mathrm{j}2\pi k\alpha/N}\mathrm{e}^{-\mathrm{j}2\pi k/N}z^{-1})$$
$$=|H(k)|\{2\cos(2\pi k\alpha/N)-[r\mathrm{e}^{-\mathrm{j}2\pi k(1+\alpha)/N}z^{-1}+r\mathrm{e}^{\mathrm{j}2\pi k(1+\alpha),N}z^{-1}]\}$$
$$=|H(k)|\{2\cos(2\pi k\alpha/N)-2r\cos[2\pi k(1+\alpha)/N]z^{-1}\} \tag{7.31}$$

合并7.30式和7.31式，我们可将$H(z)$写为

$$H(z)=\frac{1-r^Nz^{-N}}{N}\left[\sum_{k=1}^{M}\frac{|H(k)|\{2\cos(2\pi k\alpha/N)-2r\cos[2\pi k(1+\alpha)/N]z^{-1}\}}{1-2r\cos(2\pi k/N)z^{-1}+r^2z^{-2}}+\frac{H(0)}{1-rz^{-1}}\right] \tag{7.32a}$$

N为奇数时，$M=(N-1)/2$；N为偶数时，$M=(N/2)-1$。

(2) 当N为9时，我们以18 /9 = 2 kHz的间隔对频率响应进行抽样。因此，频率抽样定义为

$$|H(k)| = 1 \quad k = 0, 1, 2$$
$$0 \quad k = 3, 4$$

在这种情况下，$\alpha = (N-1)/2 = (9-1)/2 = 4$，$r = 1$。

由 7.32a 式，并且使用上面的频率抽样值，$H(z)$变成

$$H(z) = \frac{1 - z^{-9}}{9}\left\{\frac{2|H(1)|[\cos(2\pi 4/9) - \cos(2\pi 5/9)z^{-1}]}{1 - 2\cos(2\pi/9)z^{-1} + z^{-2}}\right.$$
$$\left.+ \frac{2|H(2)|[\cos(2\pi \times 2 \times 4/9) - \cos(2\pi \times 2 \times 5/9)z^{-1}]}{1 - 2z^{-1}\cos(4\pi/9) + z^{-2}} + \frac{1}{1 - z^{-1}}\right\}$$

而 $\cos(8\pi/9) = -0.9397$，$\cos(10\pi/9) = -0.9397$，$\cos(2\pi/9) = 0.7660$，$\cos(16\pi/9) = 0.7660$，$\cos(20\pi/9) = 0.7660$ 以及 $\cos(4\pi/9) = 0.1736$。将这些值代入上式，我们得出

$$H(z) = \frac{1 - z^{-9}}{9}\left[\frac{2(-0.9397 + 0.9397z^{-1})}{1 - 2 \times 0.7660z^{-1} + z^{-2}}\right.$$
$$\left.+ \frac{2(0.7660 - 0.7660z^{-1})}{1 - 2 \times 0.1736z^{-1} + z^{-2}} + \frac{1}{1 - z^{-1}}\right]$$
$$= \frac{1 - z^{-9}}{9}\left[\frac{-1.8794(1 - z^{-1})}{1 - 1.5320z^{-1} + z^{-2}} + \frac{1.5320(1 - z^{-1})}{1 - 0.3472z^{-1} + z^{-2}} + \frac{1}{1 - z^{-1}}\right]$$

图 7.23 给出了实现图。直接的和频率抽样滤波器的计算复杂性概括如下：

	加法次数	减法次数	存储
直接	8	9	18
频率抽样	10	7	25

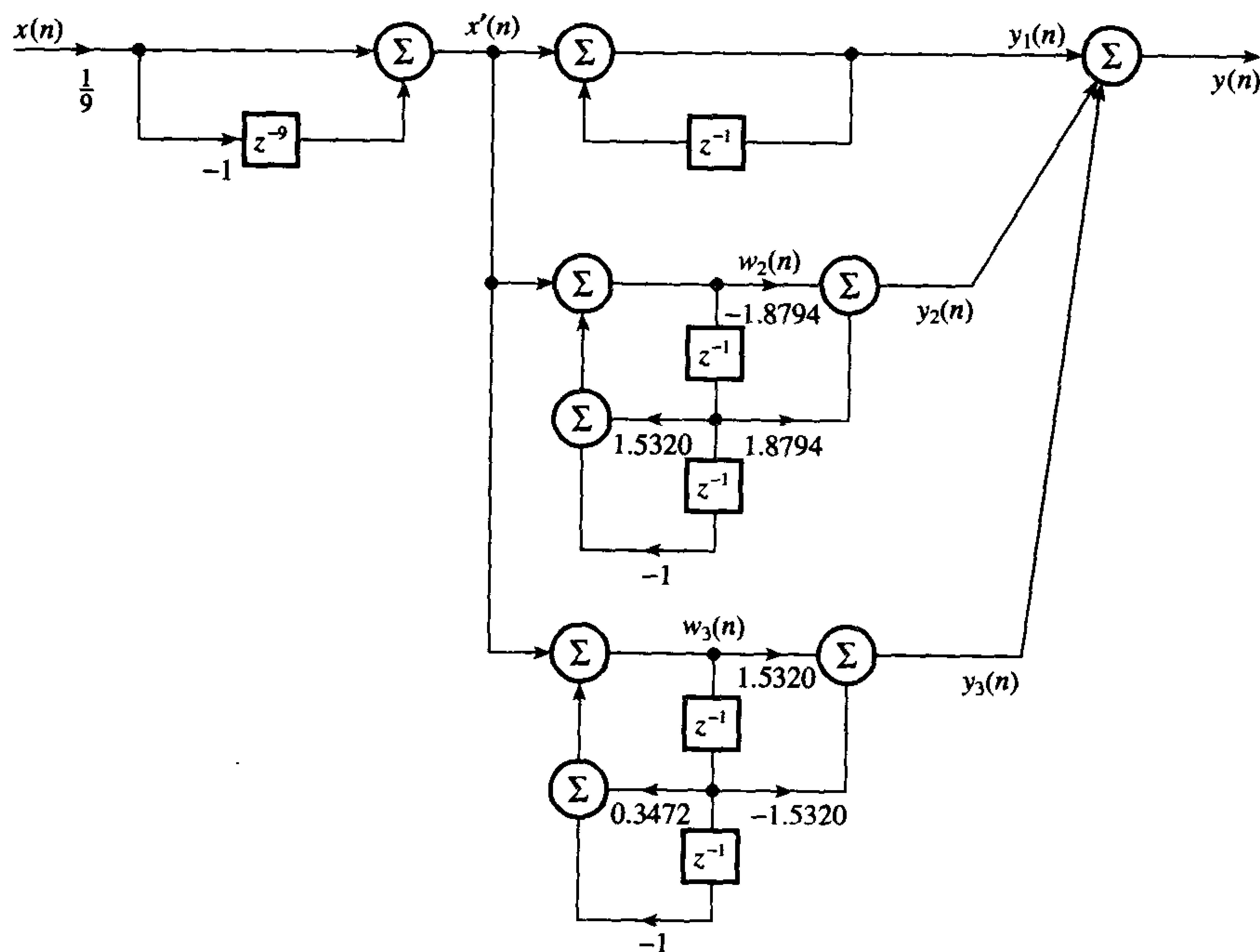


图 7.23　例 7.12 的频率抽样滤波器的实现框图

考虑图 7.23，差分方程为

$$
\begin{aligned}
x'(n) &= (1/9)[x(n) - x(n-9)] \\
y_1(n) &= x'(n) + y(n-1) \\
w_2(n) &= 1.5320 w_2(n-1) - w_2(n-2) + x'(n) \\
y_2(n) &= -1.8794 w_2(n) + 1.8794 w_2(n-1) \\
w_3(n) &= 0.3472 w_3(n-1) - w_3(n-2) + x'(n) \\
y_3(n) &= 1.5320 w_3(n) - 1.5320 w_3(n-1) \\
y(n) &= y_1(n) + y_2(n) + y_3(n)
\end{aligned} \tag{7.32b}
$$

例 7.13　求下面情况下滤波器的传递函数和差分方程：

(1) 满足下面规范的具有简单整数系数的递归 FIR 低通滤波器：

中心频率	0 Hz
抽样频率	18 kHz

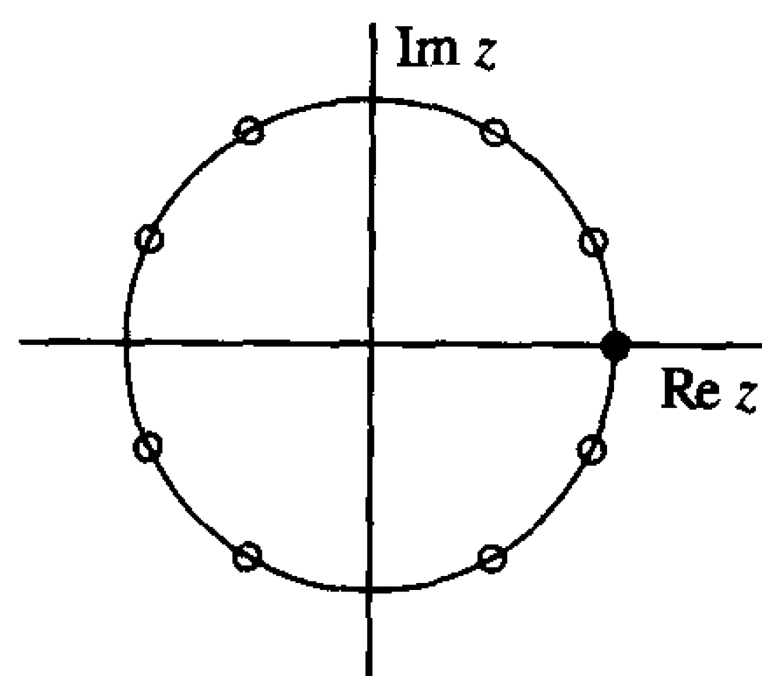
(2) 满足下面规范的具有简单整数系数的递归 FIR 带通滤波器：

中心频率	3 kHz
抽样频率	12 kHz

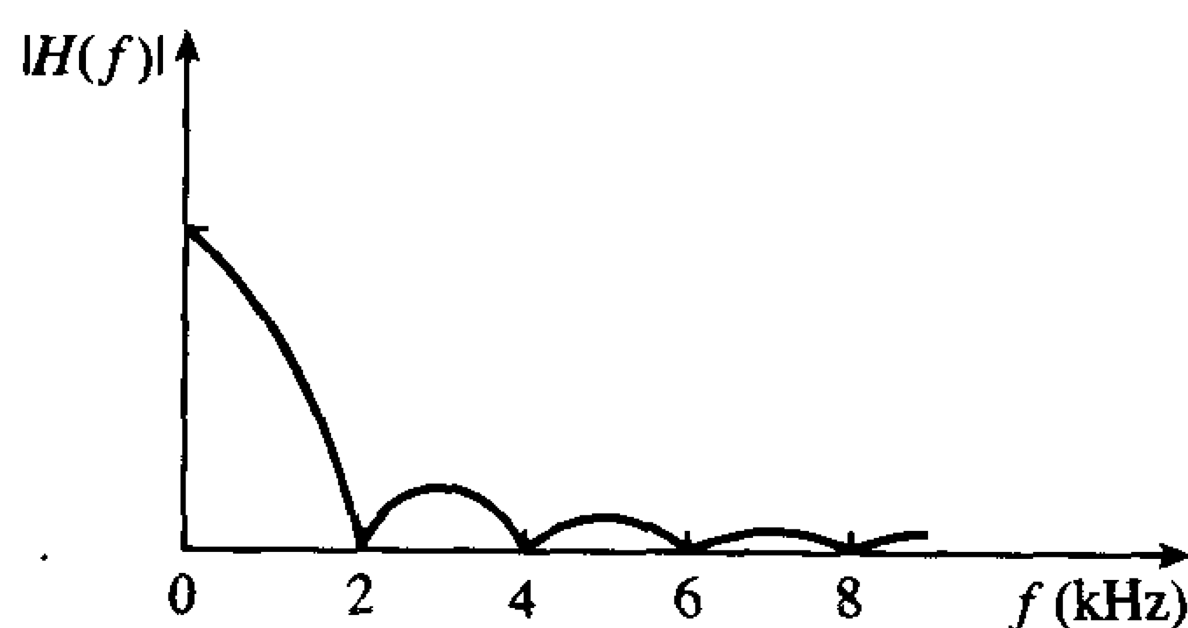
解：

(1) 若设 $N = 9$，那么频率抽样间隔为 18/9 = 2 kHz。图 7.24(a)为该情况下的极零图，图 7.24(b)为其相应的幅度响应图。根据图 7.24(a)，传递函数为

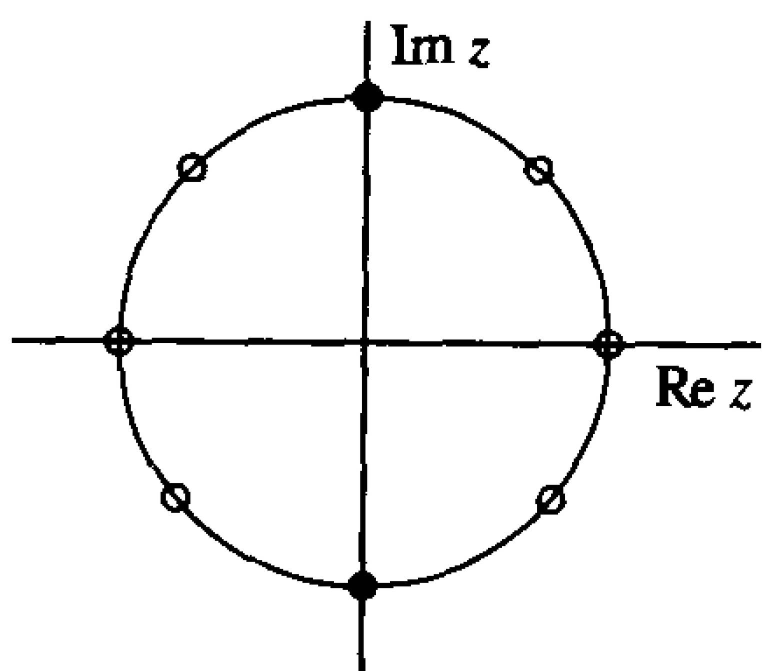
$$H(z) = \frac{1 - z^{-9}}{9} \frac{1}{1 - z^{-1}}$$



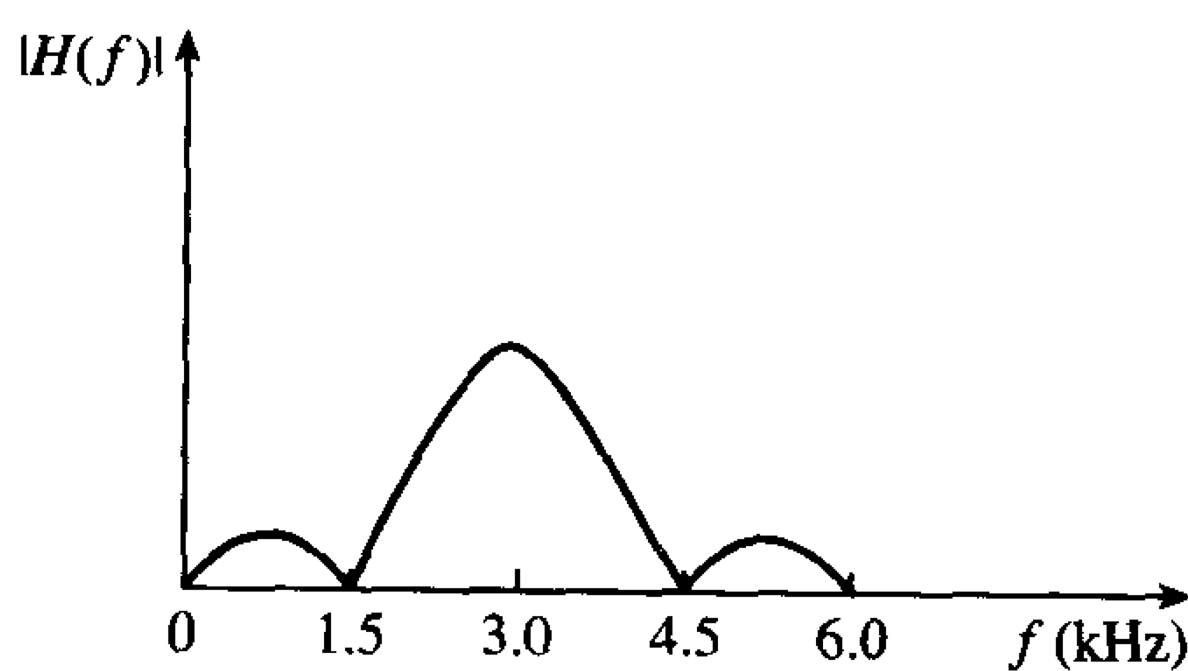
(a) 极零图



(b) 具有整数系数的递归FIR滤波器的幅度响应的概略图



(c) 具有整数系数的简单带通滤波器的极零图



(d) 相应的幅度响应图

图 7.24　极零图

对应的差分方程为

$$y(n) = y(n-1) + (1/9)[x(n) - x(n-9)]$$

(2) 由于通带以 3 kHz 为中心，所以我们必须仔细选择抽样时间来确保这种情况。假设 $N=8$，图 7.24(c)和图 7.24(d)分别给出了一个可能网格的 z 平面图和相应的幅度响应图。

$N=8$ 时，抽样间隔为 12/8 = 1.5 kHz。传递函数为

$$H(z) = \frac{1-z^{-8}}{8}\frac{1}{1+z^{-2}}$$

相应的差分方程为

$$y(n) = -y(n-2) + (1/8)[x(n) - x(n-8)]$$

显然，确定具有整数系数的频率抽样滤波器的传递函数是个非常简单的过程。然而，这种滤波器的幅度响应常常很差，而且设计者受到通带位置的限制。为改善滤波器的衰减和截止频率特性，可将传递函数提升到整数值（Lynn, 1973, 1975）。

7.7.4　频率抽样方法总结

- **第一步**　指定目标滤波器的理想的或者期望的频率响应、阻带衰减和带沿频率。
- **第二步**　根据规范：选择类型 1 频率抽样滤波器，其中以间隔 kF_s/N 取频率抽样值；或选择类型 2 的频率抽样滤波器，以间隔$(k+1/2)F_s/N$ 取频率抽样值。
- **第三步**　利用第一步中的规范和设计表（Rabiner et al., 1970），确定理想频率响应的频率抽样数 N、过渡带频率抽样数 M、通带中频率抽样数 BW、过渡带频率抽样值 T_i（$i=1,2,\ldots,M$）。
- **第四步**　用合适的公式计算滤波器系数。

此外，可以应用采用遗传算法的基于计算机的程序来执行第二步到第四步（Harris and Ifeachor, 1998）。

7.8　窗口方法、最佳方法和频率抽样方法的比较

最佳方法提供了一种简单而有效的计算FIR滤波器系数的方法。虽然最佳方法提供了滤波器规范的整体控制，但是最佳滤波器设计软件必须具有实用性。对于大多数应用来说，对于某个合理的N值，最佳方法将得到具有良好幅度响应特性的滤波器。最佳方法尤其适合设计希尔伯特变换器和差分器。其他方法对于设计希尔伯特变换器和差分器要比最佳方法产生更大的近似误差。

在缺乏最佳软件的情况下或者当通带和阻带波纹相等的情况下，窗口方法不失为一个良好选择。窗口方法是一个应用特别简单而且容易理解的方法。然而，根据滤波器系数数目，最佳方法往往给出更经济的解。窗口方法不允许设计者精确控制截止频率以及通带和阻带的波纹。

频率抽样方法是惟一允许FIR滤波器递归和非递归实现的方法。由于递归方法计算较经济，在实现时需要考虑递归实现时应该采用频率抽样方法。仅当基本算术和编程的简单性（例如标准微处理器中的汇编语言）很重要时才考虑整数系数这样的一种特殊形式。但必须做一定的检查，看看差的幅度响应是否可以接受。使用频率抽样方法可以很容易地设计出具有任何幅度相位响应的滤波器。频率抽样方法缺乏对带沿频率和通带波纹位置的精确控制，并依赖于 Rabiner et al.(1970) 的设计表的有效性（虽然存在一个基于 PC 的设计程序（Harris and Ifeachor, 1998））。

例 7.14　两个线性相位 FIR 带通滤波器要求满足下面的规范：

滤波器 1，

通带	8 ~ 12 kHz
阻带波纹	0.001
峰值通带波纹	0.001
抽样频率	44.14 kHz
过渡带宽	3 kHz

滤波器 2，

通带	8 ~ 12 kHz
阻带波纹	0.001
峰值通带波纹	0.01
抽样频率	44.14 kHz
过渡带宽	3 kHz

求并且用下面方法比较每个滤波器的频率响应。

(1) 窗口方法
(2) 频率抽样方法
(3) 最佳方法

解：

(1) **窗口方法** 对于滤波器 1，根据规范通带波纹为 20 log(1+0.001) = 0.008 68 dB，阻带衰减为 −20 log(0.001) = 60 dB。根据 7.10 式和 7.11 式，凯塞窗参数为

截止频率	6.5 kHz, 13.5 kHz
波纹参数 β	5.653
滤波器系数数目	53
抽样频率	44.14 kHz

对于滤波器2，由于窗口方法中通带和阻带波纹总是近似相等，所以结果和滤波器1的结果是一样的。图 7.25(a)给出了最终滤波器的频谱。

(2) **频率抽样方法** 对于滤波器 1，我们假定类型 1 抽样滤波器，且滤波器长度 N 选择为 53，这些都与窗口方法的相同。根据设计表（Rabiner et al., 1970），在 F_s = 44.14 kHz、M = 2、N = 53 时，发现我们需要两个过渡带频率抽样值才能实现 60 dB 的期望阻带衰减。对于 N = 53，理想频率响应的抽样值为

$$
|H(k)| = \begin{cases} 0 & k = 0, 1, \ldots, 7 \\ 0.106\,89 & k = 8 \\ 0.592\,53 & k = 9 \\ 1 & k = 10\text{–}14 \\ 0.592\,53 & k = 15 \\ 0.106\,89 & k = 16 \\ 0 & k = 17\text{–}26 \end{cases}
$$

使用程序fresamp.c（参见附录）可求得滤波器，图7.25(b)给出了对应的滤波器的频率响应。由于滤波器 1 和滤波器 2 的阻带衰减相同，那么滤波器 1 和滤波器 2 是一样的。

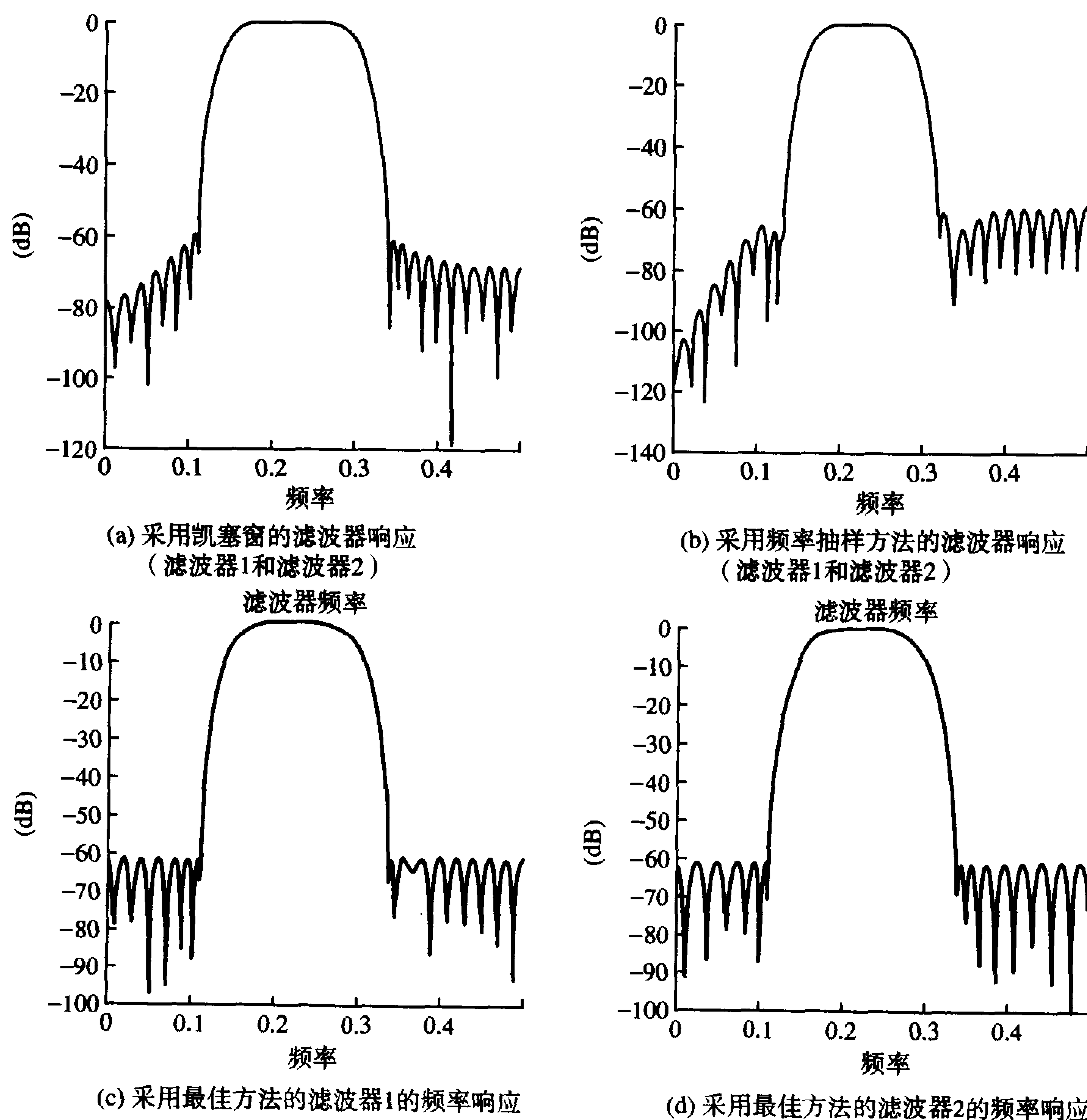
(3) **最佳方法**　对滤波器 1，根据规范，归一化的带沿频率是 0, 5/44.14、8/44.14、12/44.14、15/44.14 和 22.07/44.14，即 0、0.113 28、0.181 24、0.271 86、0.339 83 和 0.5。使用附录中的程序，我们求得 $N = 49.6$。由于通带波纹与阻带波纹相等，所以在三个频带的权值相同。最佳设计程序的输入参数为

滤波器系统数目	49
带沿频率	0, 0.113 28, 0.181 24, 0.271 86, 0.339 83, 0.5
权值	5, 5, 5

对于滤波器 2 来说，输入参数为

滤波器系数数目	39 (39.45)
带沿频率	0, 0.113 28, 0.181 24, 0.271 86, 0.339 83, 0.5
权值	10, 1, 10

图 7.25(c)和图 7.25(d)描述了采用最佳方法得到的滤波器频率响应。



(a) 采用凯塞窗的滤波器响应（滤波器1和滤波器2）

(b) 采用频率抽样方法的滤波器响应（滤波器1和滤波器2）

(c) 采用最佳方法的滤波器1的频率响应

(d) 采用最佳方法的滤波器2的频率响应

图 7.25　采用窗口方法、频率抽样方法、最佳方法的滤波器频率响应的比较

7.9 特殊 FIR 设计主题

7.9.1 半带 FIR 滤波器

半带（half-band）滤波器是FIR滤波器的一种特殊类型。半带滤波器主要吸引人的特征是它的近半数滤波器的系数值为0，由此也使计算量减少了1倍。这一特征使得半带滤波器在应用中很有意义，如多速率处理。在多速率处理需要有效地抗混叠或者抗像频，以便改变数据的抽样率（详细内容请参见第9章）。

因果的半带滤波器具有下列特征：

(1) 通带和阻带波纹相等，即

$$\delta_p = \delta_s = \delta \tag{7.33}$$

(2) 通带和阻带带沿频率有下列关系：

$$f_s = \frac{F_s}{2} - f_p \tag{7.34}$$

(3) 频率响应关于四分之一抽样频率对称。即关于 $f = F_s/4$ 对称，

$$H\left(\frac{F_s}{4} + f\right) = 1 - H\left(\frac{F_s}{4} - f\right) \tag{7.35}$$

而且在该频率处，归一化频率响应将降低一半，即

$$|H(f)| = 0.5 \left(\text{at } f = \frac{F_s}{4}\right)$$

(4) 在单位冲激响应中，当 N 为奇数时，除 $h((N-1)/2)$ 之外，系数每隔一个为0：

$$\begin{aligned} h(2n) &= 0, \qquad n = 0, 1, \ldots, (N-1)/4 \\ &\;\;\; 0.5, \qquad n = (N-1)/2 \end{aligned} \tag{7.36}$$

半带滤波器系数可以用前面描述的FIR方法求得，如窗口方法和最佳方法。在使用这些方法时，7.33式和7.34式给出的系数必须满足。

例 7.15 使用窗口方法求满足下列规范的FIR低通滤波器的系数：

通带带沿频率	2 kHz
过渡带宽	0.5 kHz
阻带衰减	>50 dB
抽样频率	8 kHz

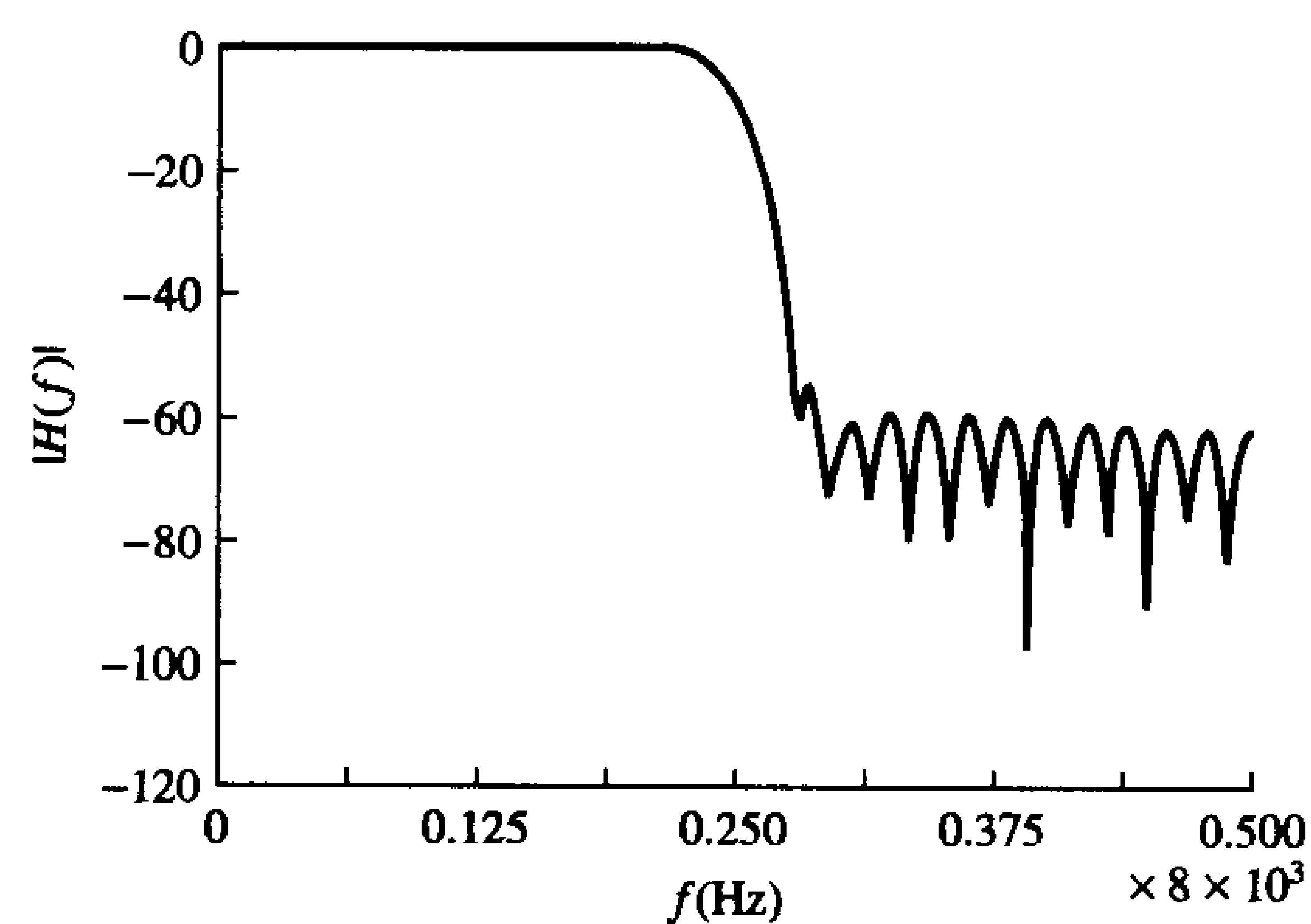
解：

表7.13中列出了滤波器系数，图7.26(a)给出了滤波器的频谱图。从表7.13中可以看出，滤波器系数从 $h(0)$ 开始每隔一个就为0（忽略由于计算 $h(n)$ 时的数值不精确的误差），系数 $h(26)$ 除外。这意味着在滤波期间，输入的抽样数据每隔一个就可以忽略一个（这样抽样频率就降低为原来的1/2）。

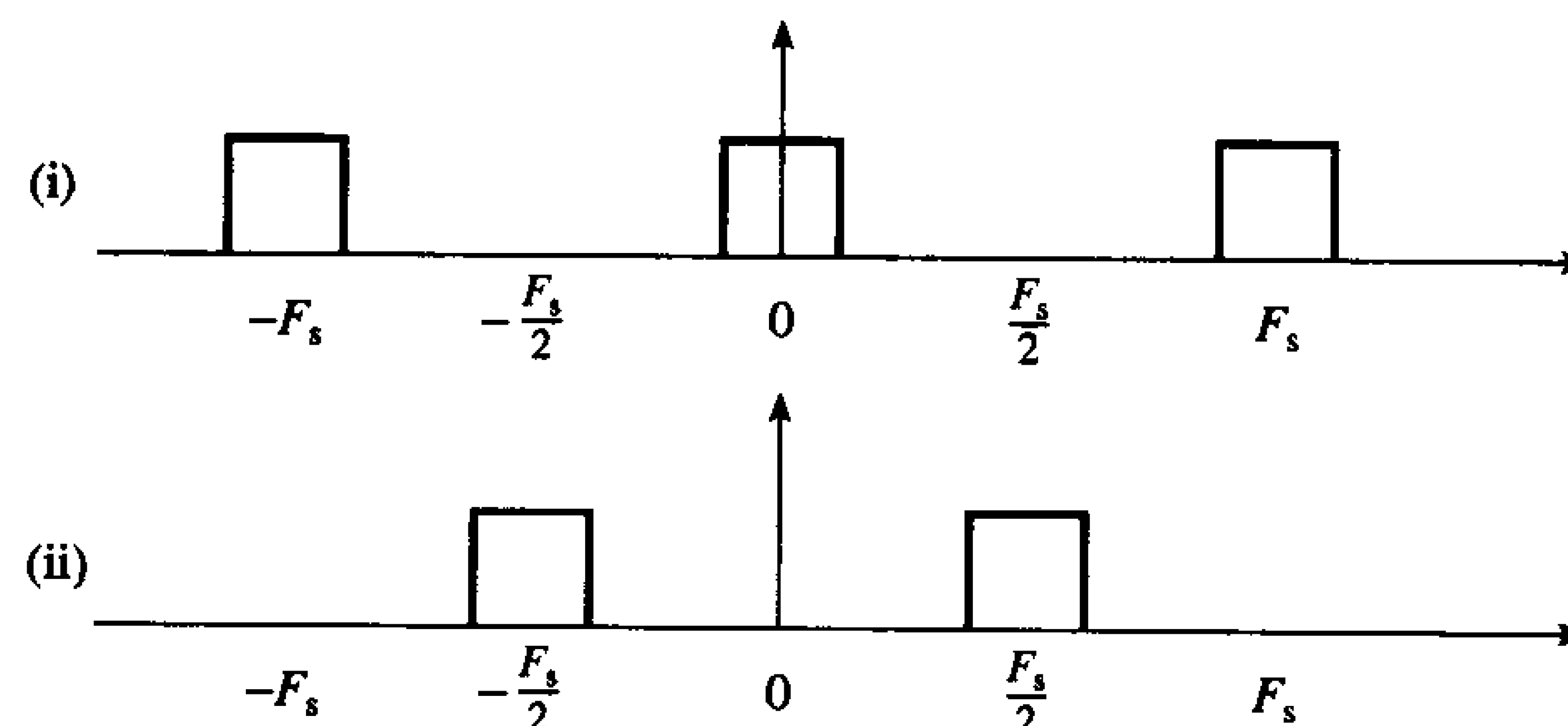
表 7.13　半带低通滤波器的系数表（哈明窗，$N=53$, $f_c=2000$ Hz）

h[0] =	−1.1243421e−09	= h[52]
h[1] =	1.1109516e−03	= h[51]
h[2] =	1.3921496e−09	= h[50]
h[3] =	−1.6473646e−03	= h[49]
h[4] =	−2.0024685e−09	= h[48]
h[5] =	2.6429869e−03	= h[47]
h[6] =	2.9211490e−09	= h[46]
h[7] =	−4.1909615e−03	= h[45]
h[8] =	−4.0967870e−09	= h[44]
h[9] =	6.4068290e−03	= h[43]
h[10] =	5.4636006e−09	= h[42]
h[11] =	−9.4484947e−03	= h[41]
h[12] =	−6.9451110e−09	= h[40]
h[13] =	1.3555871e−02	= h[39]
h[14] =	8.4584215e−09	= h[38]
h[15] =	−1.9134767e−02	= h[37]
h[16] =	−9.9188559e−09	= h[36]
h[17] =	2.6953222e−02	= h[35]
h[18] =	1.1244697e−08	= h[34]
h[19] =	−3.8674295e−02	= h[33]
h[20] =	−1.2361758e−08	= h[32]
h[21] =	5.8666205e−02	= h[31]
h[22] =	1.3207536e−08	= h[30]
h[23] =	−1.0304890e−01	= h[29]
h[24] =	−1.3734705e−08	= h[28]
h[25] =	3.1728215e−01	= h[27]
h[26] =	5.0000000e−01	= h[26]

注意每隔一个就为 0（由于数值误差不能精确为 0）。



(a) 半带低通滤波器的频率响应



(b) (i) 理想低通滤波器的频率响应
(ii) 等效的理想高通滤波器的频率响应

图 7.26　滤波器的频谱图

7.9.2 频率变换

在某些实时应用中，滤波器特性需要某些变化，无论是低通滤波器还是等效的高通滤波器。低通滤波器与高通滤波器之间存在的简单关系允许这样的变化。FIR高通滤波器系数可简单地通过改变等效低通滤波器系数值的符号而求得：

$$h_{\text{hp}}(n) = (-1)^n h_{\text{lp}}(n) \tag{7.37}$$

这个关系式是基于这样的事实，高通滤波器的频率响应与低通滤波器的频率响应经过平移半个抽样频率后是相同（参见图7.26(b)所示）。因此，高通滤波器频率响应可以通过低通滤波器频率响应用 $F_s/2-f$ 代替 f 来求得：

$$H_{\text{hp}}(f) = H_{\text{lp}}\left(\frac{F_s}{2} - f\right) \tag{7.38}$$

例 7.16 低通滤波器的特性如下：

通带带沿频率	1.5 kHz
抽样频率	10 kHz
滤波器系数数目	15

(1) 采用哈明窗求低通滤波器的系数值。

(2) 写下等效高通滤波器的规范，并由这些规范求高通滤波器的系数值。

(3) 由上述变换公式求等效高通滤波器的系数值。

解：

(1) 将上述参量作为程序 window.c 的输入值，得到表 7.14 中的系数值。

(2) 等效高通滤波器的规范：

通带带沿频率	$F_s/2-f_c = 5000 - 1500$ kHz = 3500 kHz
抽样频率	10 kHz
滤波器系数数目	15

将这些参量作为设计程序 window.c 的输入，求得表 7.14 所列的高通滤波器的系数值。

(3) 应用上面的简单变换公式，求出高通滤波器的系数值，与(2)中得到的系数值是相等的。

表 7.14 低通滤波器和等效的高通滤波器的系数值

	低通	高通
$h(0)$	1.2654×10^{-3}	1.2654×10^{-3}
$h(1)$	-5.2341×10^{-3}	5.2341×10^{-3}
$h(2)$	-1.9735×10^{-2}	-1.9735×10^{-3}
$h(3)$	-2.3009×10^{-2}	2.3009×10^{-3}
$h(4)$	2.2366×10^{-2}	2.2366×10^{-2}
$h(5)$	1.2833×10^{-1}	-1.2833×10^{-1}
$h(6)$	2.4728×10^{-1}	2.4728×10^{-1}
$h(7)$	3.0000×10^{-1}	-3.0000×10^{-1}

7.9.3 FIR 滤波器的计算效率

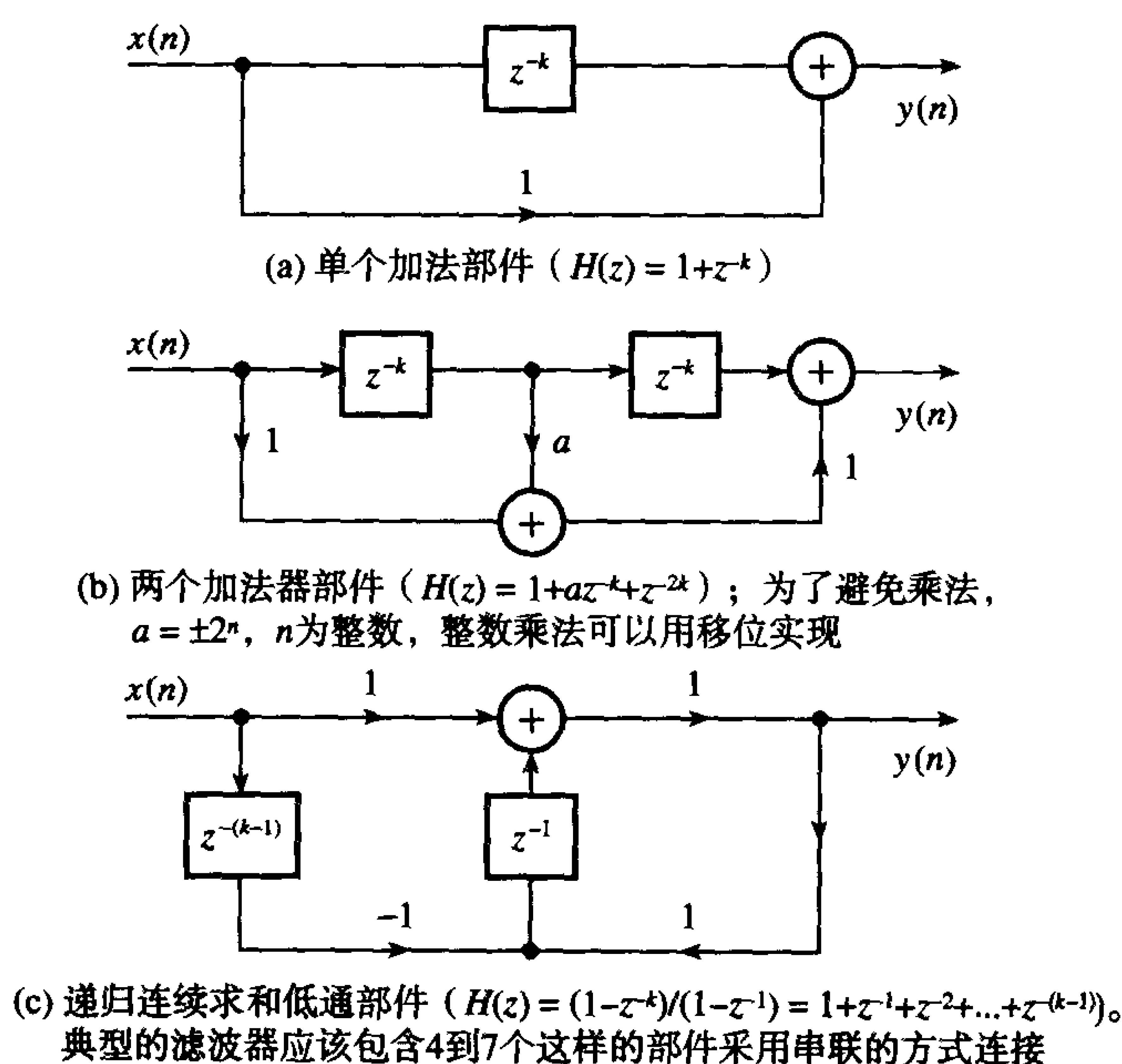
在一些应用中，本章描述的方法并不一定适用。例如，一些应用中由线性相位FIR滤波器引入的相位延迟可能长得让人不能接受（如类型1的FIR滤波器的相位延迟是$(N-1)T/2$，比N大）。例如

在控制系统，反馈环路里使用这样的滤波器会导致系统不稳定。而在这种情况下，比较适合使用最小相位滤波器（Parks and Burrus, 1987）。

最佳方法的等波纹特征会导致滤波器冲激响应中产生回波（echo），这样的效果是人们不希望看到的，一个平滑的频率响应特性可以降低冲激响应尾部上的回波。

在其他一些应用中，如图像处理，当使用的是标准 FIR 滤波器时，它的算术运算数可能太大。遗憾的是整数系数滤波器不适合这些应用，因为它的幅度响应特性较差。FIR 滤波器只要求非常简单的算术操作，但就其幅度响应与标准 FIR 滤波器的幅度响应可以相比拟，这样的 FIR 滤波器更适合于这些应用（Wade et al., 1990; Mitra and Kaiser, 1993）。

这种方法的基础是将两个或多个基本滤波器部件串联起来，这些基本的部件如图7.27所示。几乎每一个基本部件都不包含任何乘法运算。



(a) 单个加法部件（$H(z)=1+z^{-k}$）

(b) 两个加法器部件（$H(z)=1+az^{-k}+z^{-2k}$）；为了避免乘法，$a=\pm 2^n$，n为整数，整数乘法可以用移位实现

(c) 递归连续求和低通部件（$H(z)=(1-z^{-k})/(1-z^{-1})=1+z^{-1}+z^{-2}+\ldots+z^{-(k-1)}$）。典型的滤波器应该包含4到7个这样的部件采用串联的方式连接

图 7.27　基本的 FIR 滤波器部件的例子

该方法的主要问题是，要找到一种有效方法选择要串联的基本滤波器部件是很困难的；事实上仅能有效地设计低阶滤波器。人们已经采用遗传算法来解决此类问题（Suckley, 1990）。

7.10　FIR 滤波器的实现结构

FIR 滤波器的特征可以通过下面给出的传递函数 $H(z)$来刻画：

$$H(z)=\sum_{n=0}^{N-1}h(n)z^{-n}$$

实现结构本质上是传递函数理论上等效的框图表示。在多数情况下，它们是由乘法器、加法器/求和器以及延迟单元组成的。FIR 滤波器实现结构有许多种，这里仅讨论常用的实现结构。

7.10.1　横向结构

图 7.28 描述了横向（或节拍延迟）结构。对于这种结构，滤波器的输入 $x(n)$与输出 $y(n)$之间的关系式为

$$y(n)=\sum_{m=0}^{N-1}h(m)x(n-m) \tag{7.39}$$

图中，符号z^{-1}代表一个抽样延迟或单位时间延迟。那么$x(n-1)$是$x(n)$延迟一个抽样的值。在数字实现中，标有z^{-1}的方框可能表示移位寄存器或者通常的RAM中的存储位置。横向滤波器结构是最流行的FIR滤波器结构。

输出抽样值$y(n)$是当前输入$x(n)$和前面$N-1$个输入抽样值的加权和，即$x(n-1)$到$x(n-N)$的加权和。对于横向结构，每一个输出抽样值$y(n)$的计算要求：

- $N-1$个存储位置来保存$N-1$个输入抽样值
- N个存储位置来保存N个系数
- N个乘法器
- $N-1$个加法器

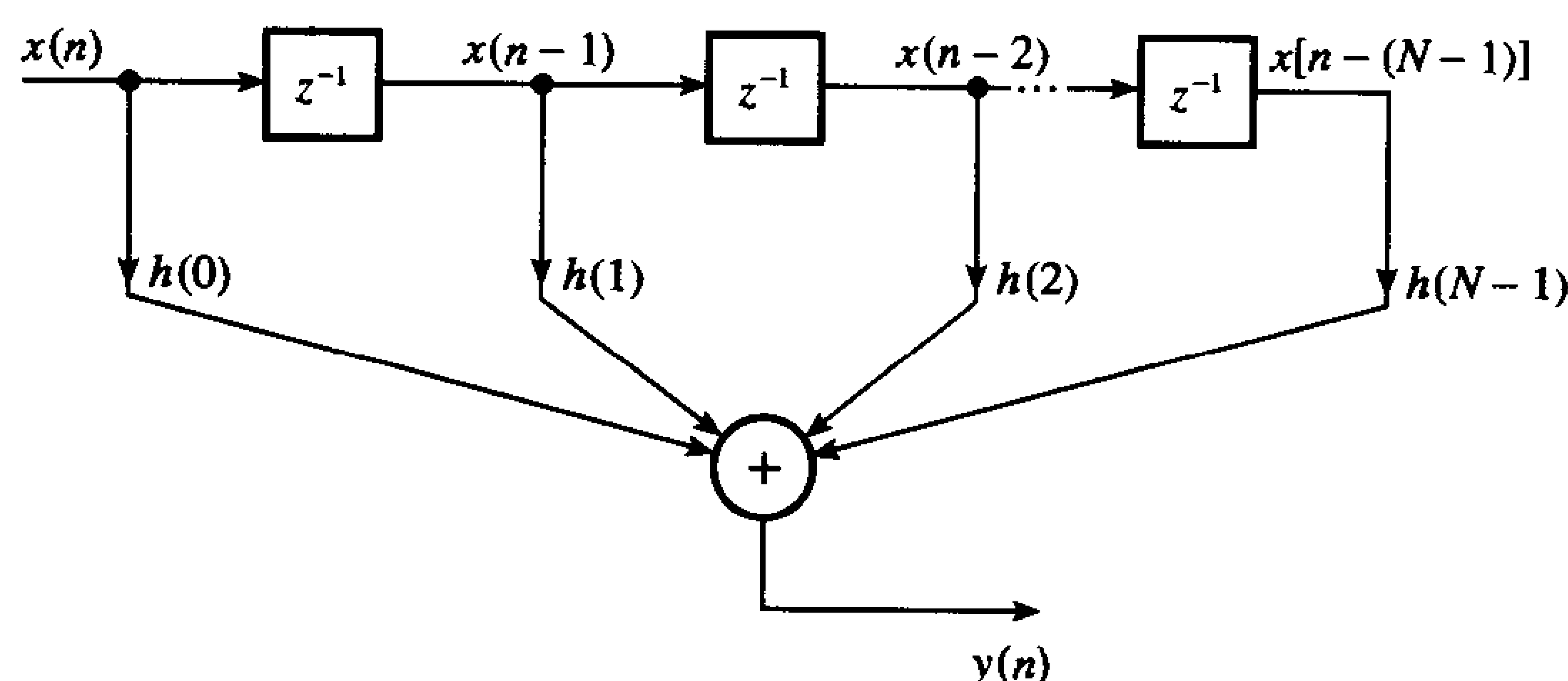


图 7.28　横向滤波器结构

7.10.2　线性相位结构

线性相位结构是横向结构的演变形式，它充分利用线性相位FIR滤波器的冲激响应系数对称性的优点来降低滤波器实现的计算复杂性。

在线性相位滤波器中，它的系数是对称的，即$h(n)=\pm h(N-n-1)$。因此，考虑到这种对称性，滤波器方程可以重写，这样加法和减法运算量都将降低。对于类型1和类型2的线性相位滤波器，滤波器的传递函数可写为

$$H(z)=\sum_{n=0}^{(N-1)/2-1}h(n)[z^{-n}+z^{-(N-1-n)}]+h\left(\frac{N-1}{2}\right)z^{-(N-1)/2}\quad N\text{为奇数} \tag{7.40a}$$

$$H(z)=\sum_{n=0}^{N/2-1}h(n)[z^{-n}+z^{-(N-1-n)}]\quad N\text{为偶数} \tag{7.40b}$$

相应的差分方程为

$$y(n)=\sum_{k=0}^{(N-1)/2-1}h(k)\{x(n-k)+x[n-(N-1-k)]\}+h[(N-1)/2]x[n-(N-1)/2] \tag{7.41a}$$

$$y(n)=\sum_{k=0}^{(N-1)/2-1}h(k)\{x(n-k)+x[n-(N-1-k)]\} \tag{7.41b}$$

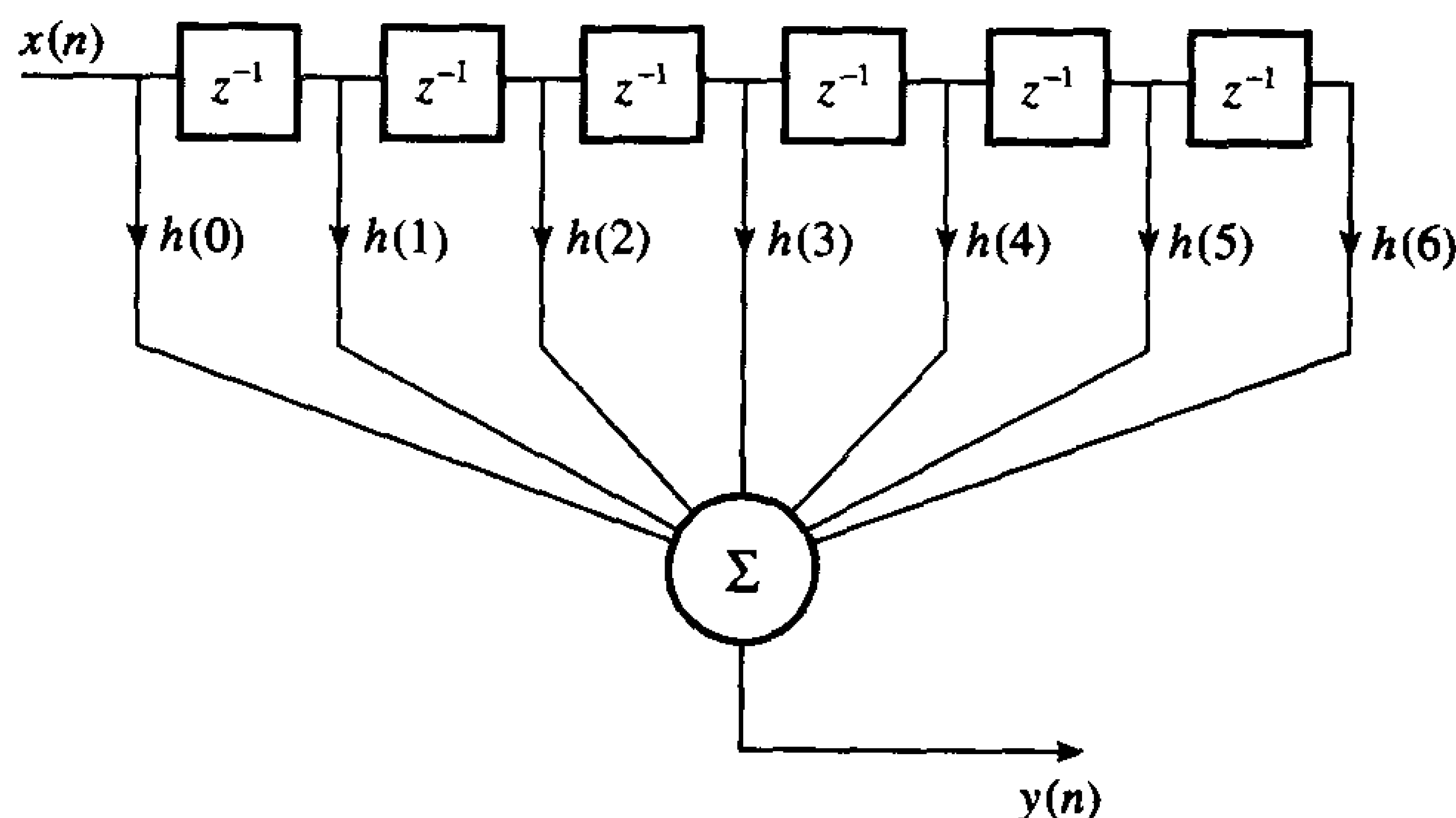
7.39式和7.41式的比较表明线性相位滤波器计算更为有效，要求的加法和乘法运算只有一半。然而，在许多DSP处理器中，7.39式的实现更有效。这是因为在7.41式中，数据隐含许多复杂的索引而使其计算的优势丧失。

例7.17　线性相位FIR滤波器有7个系数，列出如下。画出滤波器实现图，(a)用直接（横向）结构，(b)用线性相位结构。比较它们的计算复杂度。

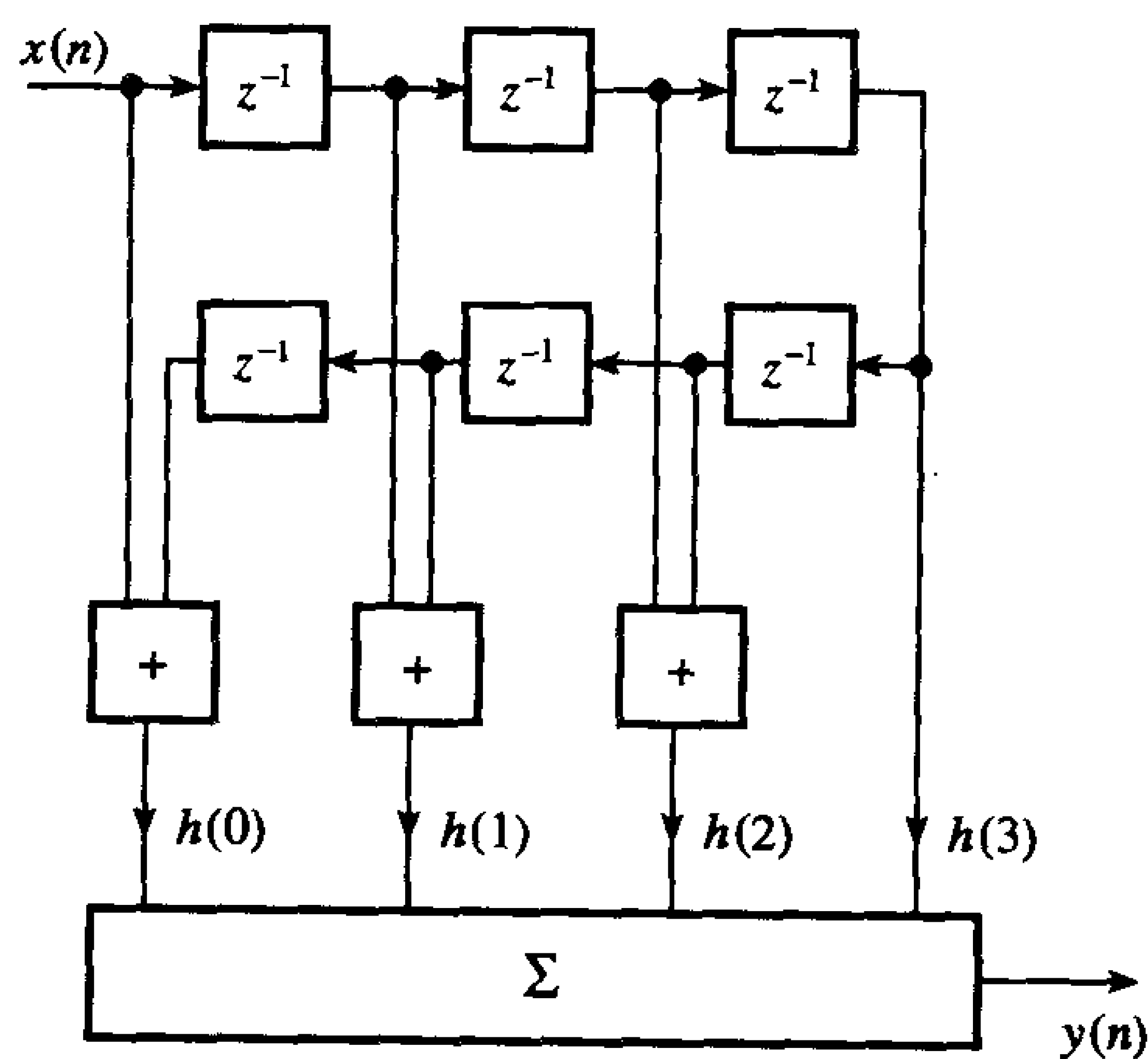
$$h(0)=h(6)=-0.032$$
$$h(1)=h(5)=\ 0.038$$
$$h(2)=h(4)=\ 0.048$$
$$h(3)\qquad\quad=-0.048$$

解：

实现图如图7.29所示。



(a)



(b)

图7.29　例7.17的(a)横向结构和(b)线性相位结构

7.10.3　其他结构

7.10.3.1　快速卷积

快速卷积方法包括在频率域计算7.39式的卷积运算。正如我们在第5章中所讨论的那样，时域卷积等价于频域相乘。简而言之，这里的滤波是首先通过进行$x(n)$和$h(n)$的傅里叶变换，然后将它们相乘再做反变换而得到的。图7.30描述了这一概念。实际上，在实时滤波中用到了所谓的重叠相加和重叠保留技术，这些技术已经在第5章讨论了。

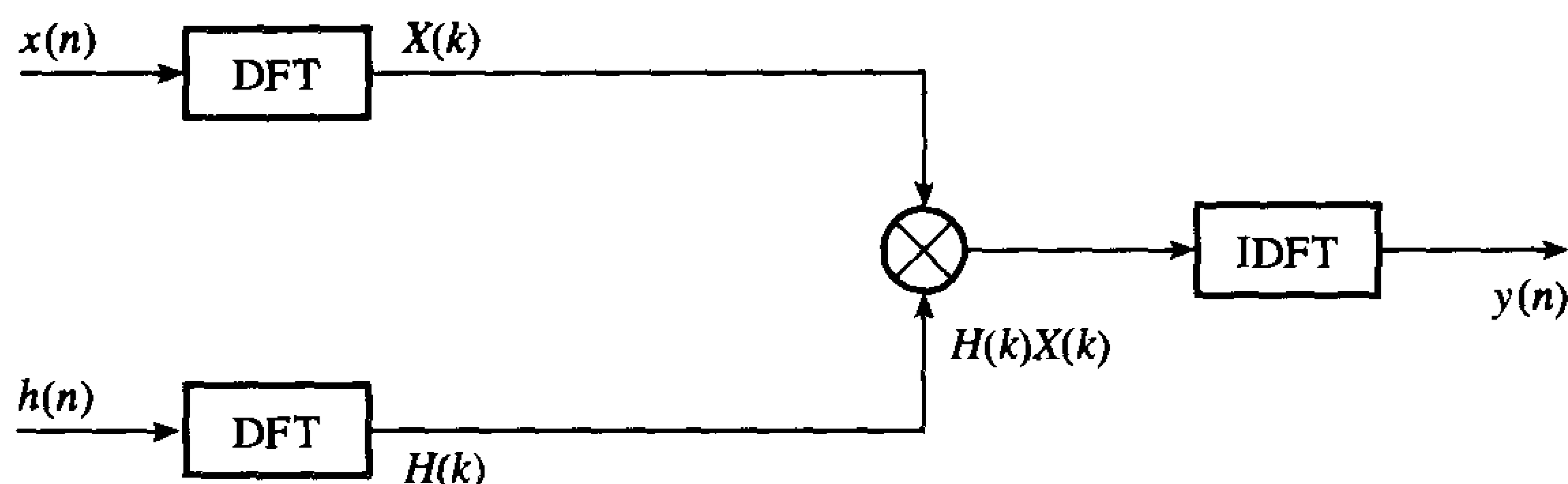


图 7.30 快速卷积实例图

7.10.3.2 频率抽样结构

在频率抽样结构中，滤波器是由期望的频率响应的抽样值$H(k)$而不是滤波器冲激响应的系数来刻画的。这种情况我们已经详细讨论过了。对于窄带滤波器，大多频率抽样值为0。所以，得到的频率抽样滤波器比等效的横向结构的滤波器需要的系数更少。因此，需要的乘法和加法也少。图 7.22 给出了其典型的实现图。

7.10.3.3 转置和串联结构

除了部分和加入到后级之外，转置结构与直接结构是相似的。这种方法比直接法更易于受到舍入噪声的影响。在串联结构的实现中，传递函数$H(z)$表示为二阶和一阶部分的乘积。在当前的DSP中，FIR 滤波器很少使用转置和串联结构。

7.10.4 结构选择

结构的选择取决于多种因素和折中考虑，其中包含实现的容易程度，也就是软件硬件的复杂度，求冲激响应或传递函数系数的难易程度，以及它们对系数量化的相对敏感度。在实际中，系数描述的精度受限于所用处理器的字长。使用有限的位数来表示描述每个系数趋向于将零从期望的位置移出，这会导致频率响应上的偏差。响应中偏差的大小取决于位数和所用的结构。

直接结构非常易于编程，且可由大多数DSP芯片有效地实现，因为这些芯片有针对横向FIR滤波器的指令。直接构造是实现非递归滤波器的最常用方法，它的最主要的优点是其简单性，要求的器件最少，且数据的存储不复杂。串联结构对系数误差和量化噪声不敏感，但是系数值需要做更大的努力才能求出，且程序设计不适合 DSP 芯片的结构。快速卷积结构能提供出比其他方法更好的计算优势，但需要有效的 FFT。

对于窄带频率选择滤波器来说，频率抽样结构比等价的横向结构计算更有效。在此滤波器中，仅有相对少量的频率抽样值不为零，因此每输出一个值仅需非常少的乘法。然而，频率抽样结构可能要求更为复杂的编程，因为在其差分方程中固有的对数据的复杂索引（比较 7.39 式和 7.32b 式）。为了避免稳定问题，频率抽样结构的零点和极点应该位于单位圆稍内一点，如半径$r=0.99$的圆上。当要求 FIR 滤波器递归实现时，这种结构是一种自然的选择。这种结构易于模块化，并有利于并行处理。

总之，除非规范要求采用频率抽样结构，或者需要计算数据的谱采用快速卷积，那么采用横向结构不失为一个好的选择。

7.11 FIR 数字滤波器的有限字长效应

实际上，FIR 数字滤波器的实现常常是使用DSP处理器（如德州仪器的TMS320C50），为FIR滤波器或者希望高速处理的应用设计的定制算法的 DSP 芯片（如 INMOS A100），以及由乘法器、存储单元、加法器和控制器（例如 Plessey 的 PDSP1600 族）构成的构建块等。在这些情况下，用来表示加到滤波器的输入数据、滤波器的系数以及执行算术运算的位数从效率和限制数字滤波器的成

本来考虑必须尽可能小。这种由采用有限位数引起的问题称为有限字长效应，一般来说都会导致滤波器性能的下降。

在这一节我们将讨论有限字长效应对FIR数字滤波器性能的影响，并提出使这种影响最小的方法。讨论集中在直接形式的FIR结构上，因为在现代信号处理中它是最具有吸引力的FIR结构，并且采用舍入，这是一种最简单也是最为广泛使用的量化方法。

有限字长效应对 FIR 数字滤波器的影响表现在以下四个方面。

(1) **ADC 噪声**　这是一种熟悉的 ADC 量化噪声，当滤波器输入是从模拟信号中得到时引起的噪声。ADC 噪声限制了可得到的信噪比（SNR）。这种影响可以通过附加位数，使其与固有的信噪比一致来减少其影响（参见第 13 章），或通过多速率技术来提高信噪比（参见第 9 章）。

(2) **系数量化误差**　这是由用有限位数来表示滤波器系数时带来的量化误差。系数量化误差对于修改期望的频率响应有着不利的影响。例如，在滤波器阻带，系数量化误差限制了可能的最大衰减，因此允许附加的信号传输。一个简单的解决方法是使用足够多的位表示滤波器的系数。然而，最佳技术允许有效地选择系数使系数字长最小。

(3) **算术运算的量化结果带来的舍入误差**　例如，在存储乘法的结果前丢弃低位就会产生这样的舍入误差。这通常迫使我们使用处理器使用的字长。这种误差降低了信噪比，乘积之和的双倍长度可以降低这类舍入误差。误差的大小与算术运算的类型和滤波器的结构有关。

(4) **算术溢出**　当部分求和或滤波器输出超过系统所允许的字长时将发生算术溢出。实际上，当出现溢出时，输出的样本值是错误的（通常符号改变）。减少溢出或避免溢出的一种方法是对滤波器的系数做一定的伸缩，即对滤波器的每个系数除以一个因子，使滤波器输出的样本值永远也不会超出允许的字长。显然这样是以降低信噪比为代价的。

下面几节我们来对(2)至(4)进行讨论。

7.11.1　系数量化误差

任何一种近似方法得到的滤波器系数（例如窗口方法或最佳方法）通常都精确到小数点的几位。为了实现滤波器，滤波器系数必须由固定的位数表示，并且这个固定的位数常常是由使用的处理器的字长决定的。例如，如果滤波器中我们使用 16 位 DSP 处理机，则滤波器系数就由 16 位字长来表示。然而这样做就会自动地引入误差，这种误差使得有限字长滤波器的频率响应偏离期望的响应。在某些情况下，这种偏离意味着初始的规范不再满足。

例 7.18　确定下面滤波器的系数由于舍入到 8 位而引起的量化误差的影响：

阻带衰减	> 90 dB
通带波纹	< 0.002 dB
通带边沿频率	3.375 kHz
阻带边沿频率	5.625 kHz
抽样频率	20 kHz
系数的个数	45

解：

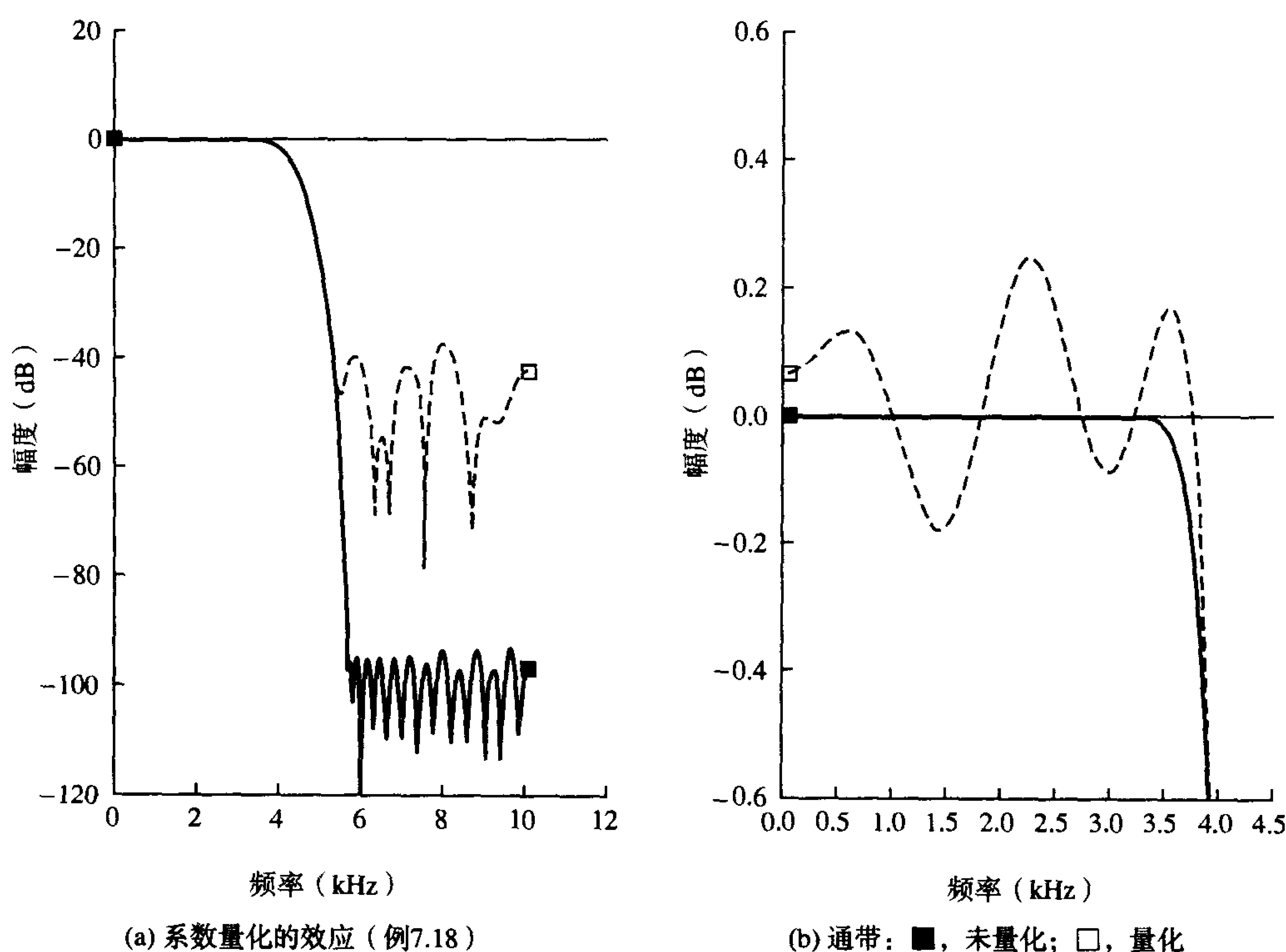
使用带有下面输入的设计程序 optimal.c：

滤波器系数数目　45
带沿频率　0, 0.168 75, 0.281 25, 0.5
权值　1, 7.28

表7.15列出了在舍入到8位前后的滤波器系数。图7.31是相应的频率响应。可以看出量化后最小阻带衰减是36 dB，性能恶化了58 dB。显然在该例中需要超过8位的字长来描述滤波器的系数。

表7.15　量化到8位前后的滤波器系数

$h(n)$	$h_q(n)$
−1.05023e−04	0.00000e+00
−1.25856e−04	0.00000e+00
3.07141e−04	0.00000e+00
6.79484e−04	0.00000e+00
−2.89029e−04	0.00000e+00
−1.77474e−03	0.00000e+00
4.08318e−04	0.00000e+00
3.43482e−03	0.00000e+00
2.66515e−03	0.00000e+00
−5.00314e−03	−7.81250e−03
−7.30591e−03	−7.81250e−03
5.09712e−03	7.81250e−03
1.48422e−02	1.56250e−02
−1.40255e−03	0.00000e+00
−2.49785e−02	−2.34375e−02
−9.39383e−03	−7.81250e−03
3.64568e−02	3.90625e−02
3.28505e−02	3.12500e−02
−4.72008e−02	−4.68750e−02
−8.52427e−02	−8.59375e−02
5.48855e−02	5.46875e−02
3.10921e−01	3.12500e−01
4.42322e−01	4.45212e−01
3.10921e−01	3.12500e−01
5.48855e−02	5.46875e−02
−8.52427e−02	−8.59375e−02
−4.72008e−02	−4.68750e−02
3.28505e−02	3.12500e−02
3.64568e−02	3.90625e−02
−9.39383e−03	−7.81250e−03
−2.49785e−02	−2.34375e−02
−1.40255e−03	0.00000e+00
1.48422e−02	1.56250e−02
5.09712e−03	7.81250e−03
−7.30591e−03	−7.81250e−03
−5.00314e−03	−7.81250e−03
2.66515e−03	0.00000e+00
3.43482e−03	0.00000e+00
4.08318e−04	0.00000e+00
−1.77474e−03	0.00000e+00
−2.89029e−04	0.00000e+00
6.79484e−04	0.00000e+00
3.07141e−04	0.00000e+00
−1.25856e−04	0.00000e+00
−1.05023e−04	0.00000e+00



(a) 系数量化的效应（例7.18）

(b) 通带：■，未量化；□，量化

图 7.31　频率响应

系数误差效应导致频率响应偏离期望响应。这种偏离在极端情况意味着规范不再满足。对一个特定的滤波器设计问题，通过求不同系数字长时滤波器的频率响应来确定合适的系数字长。从这些可以确定满足期望的规范所需要的最小位数。然而，通过分析系数量化引入的误差可加深对有限字长滤波器的设计的理解。

量化和未量化系数分别表示为 $h(n)$和 $h_q(n)$，它们的关系为

$$h_q(n) = h(n) + e(n), \quad n = 0, 1, \ldots, N-1 \tag{7.42}$$

式中 $e(n)$是量化和未量化系数之间的误差。在频域，7.42 式可以表示为

$$H_q(\omega) = H(\omega) + E(\omega) \tag{7.43}$$

式中 $E(\omega)$为期望的频率响应的误差，它可以表示为

$$E(\omega) = \sum_{m=0}^{N-1} e(m)\exp(-\mathrm{j}\omega m)$$

$H_q(\omega)$和$H(\omega)$分别是量化前后滤波器的频率响应。图7.32(a)和图7.32(b)分别给出了7.42式和7.43式的图解表示。我们可以将$e(n)$看作为与期望的滤波器并行的另一个滤波器的冲激响应（Rabiner and Gold, 1975）。系数误差在频率域上的影响可以表示为与一个非常精确的滤波器的传递函数并行的一个寄生（stray）的传递函数。设计者的目标是要限制$E(\omega)$的幅度，使实际滤波器的频率响应满足规范。

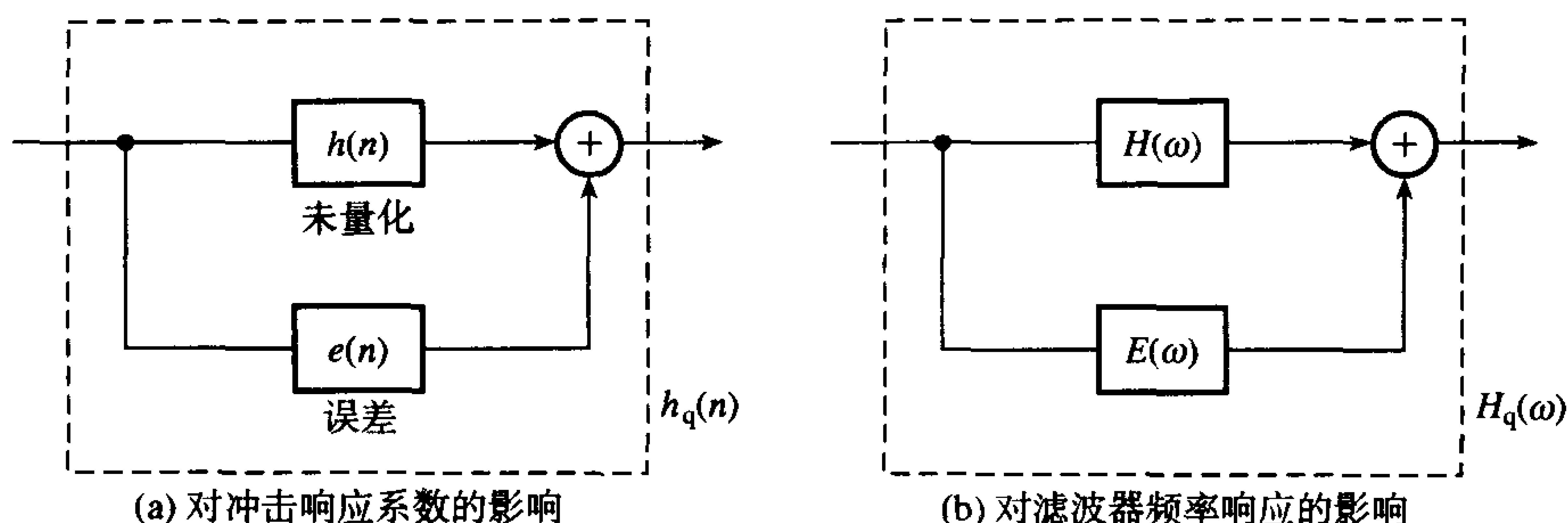


图 7.32 系数量化效应的图解

对于频率选择滤波器（低通、带通、带阻滤波器），一些研究者已确定了在频率响应中误差的边界。这些边界为给定滤波器确定合适的系数字长提供了有用的指导。当这些滤波器的精确的特性预先不知道时，这些边界对于估计自适应 FIR 滤波器要求的系数字长是有用的（参见第 10 章）。

对于直接形式的 FIR 结构，假定采用舍入，下面就是最广泛使用的边界：

$$|E(\omega)| = N2^{-B} \tag{7.44a}$$

$$|E(\omega)| = 2^{-B}(N/3)^{1/2} \tag{7.44b}$$

$$|E(\omega)| = 2^{-B}[(N\log_e N)/3]^{1/2} \tag{7.44c}$$

其中 B 是用来表示每一个系数的位数，N 是滤波器长度。边界 7.44a 式是绝对上界，这是从最坏情况的假设推出的（参见例 7.19），因此也是最悲观的。边界 7.44b 式和 7.44c 式是统计边界，它们也可以在频率响应中给出误差更精确的估计，并且估计出要使用的系数字长。统计边界假定量化误差 $e(n)$ 是均匀分布，且具有零均值。

例 7.19

(1) 阐述任何假设，证明：对于直接形式的低通 FIR 滤波器，采用舍入量化系数，可能的最大阻带衰减 A_{max} 为

$$A_{max} \leqslant 20\log_{10}(2^{-B}N) \tag{7.45}$$

(2) 低通 FIR 滤波器的技术规格：

通带偏差	0.05 dB
抽样频率	10 kHz
通带边沿	1.8 kHz
过渡带宽	500 Hz
系数的个数	65

(a) 对于一个阻带衰减至少为 60 dB 的滤波器，估计表示每个系数要求的位数。

(b) 如果在(a)中的系数字长被采用，估计期望通带波纹的增加及阻带衰减的减少，用分贝表示。

(c) 比较用(a)中得到的系数字长的滤波器的实际阻带衰减和通带波纹。

解：

(1) 由系数量化误差 $e(m)$ 定义响应 $E(\omega)$：

$$E(\omega) = \sum_{m=0}^{N-1} e(m) \exp(-\mathrm{j}\omega m)$$

其中N是滤波器长度。对于舍入量化，最坏情况的量化误差是$|e(m)| = 2^{-(B-1)}/2 = 2^{-B}$，其中$B$是系数字长（假定用2的补码表示），如果我们假定所有系数的误差都为最坏情况下的误差，那么我们可得出：

$$|E(\omega)| = \sum_{m=0}^{N-1} |e(m)| \exp(-\mathrm{j}\omega m) = \sum_{m=0}^{N-1} 2^{-B} \exp(-\mathrm{j}\omega m)$$

$$= 2^{-B} \sum_{m=0}^{N-1} \exp(-\mathrm{j}\omega m) = 2^{-B} N$$

如果$e(m)$被看作为与期望滤波器并行的另一个滤波器的冲激响应，那么在通带和阻带中的极限偏差是$2^{-B}N$，所以

$$A_{\max} < 20 \log_{10}(2^{-B}N) \text{ dB}$$

显然，这个边界是过于保守的。通常使用比建议要少的位数就足够了，然而这些边界可以作为简单应用的指导。

(2) (a) 由上述边界，设$A_{\max} = 60$ dB，$N = 65$，我们求得$B = 15.988$位。所以要求系数字长为$B = 16$位。

(b) 量化后，通带的最坏情况的峰值波纹$R_{\max}$和阻带衰减$A_{\max}$可以表示为

$$R_{\max} = 20 \log(1 + \delta_p + |E(\omega)|) = 20 \log(1 + 0.005\,773 + 0.001)$$
$$= 0.0586 \text{ dB}$$

即增加 0.0086 dB，并且

$$A_{\max} = -20 \log(\delta_s + |E(\omega)|) = -20 \log(0.001 + 0.001) = 54 \text{ dB}$$

即减少了 6 dB（δ_p和δ_s是未量化滤波器的通带和阻带偏差）。

(c) 使用最佳设计程序和下面这些参数：

系数个数	65
带沿频率	0, 0.18, 0.23, 0.5
通带－阻带权值	1, 5.773

图 7.33 显示了量化前滤波器频谱。量化（16位字长）和未量化频率响应没有很大的差别。量化前通带波纹和阻带衰减分别是0.0224 dB和66.96 dB，量化后分别是0.0227 dB和64.15 dB。

显然系数量化最主要的影响是峰值通带波纹增加和最大阻带衰减的减少。在实际过程中，在计算滤波器的系数时考虑这些影响是有用的。从本质上讲，这意味着将未量化的滤波器规范变换成一套新的规范，然后用新规范来求系数。这种变换必须是系数量化后仍满足原来的规范。

得到的滤波器可能不是最佳的。这已引起最佳技术的开发，如混合整数规划算法，来求有限字长 FIR 滤波器（例如，Lawrence and Salazar, 1980）的系数。新的方法与简单的舍入相比大大减少了系数的字长，而求合适的系数字长对中等大小的N也包含过高的计算量。一个实际的方法是使用7.44式中的边界来估计表示系数所需要的位数。需要的系数字长常常比该值或高或低1到4位，这个字长可以通过研究对应于该范围内字长的频率响应来确定。

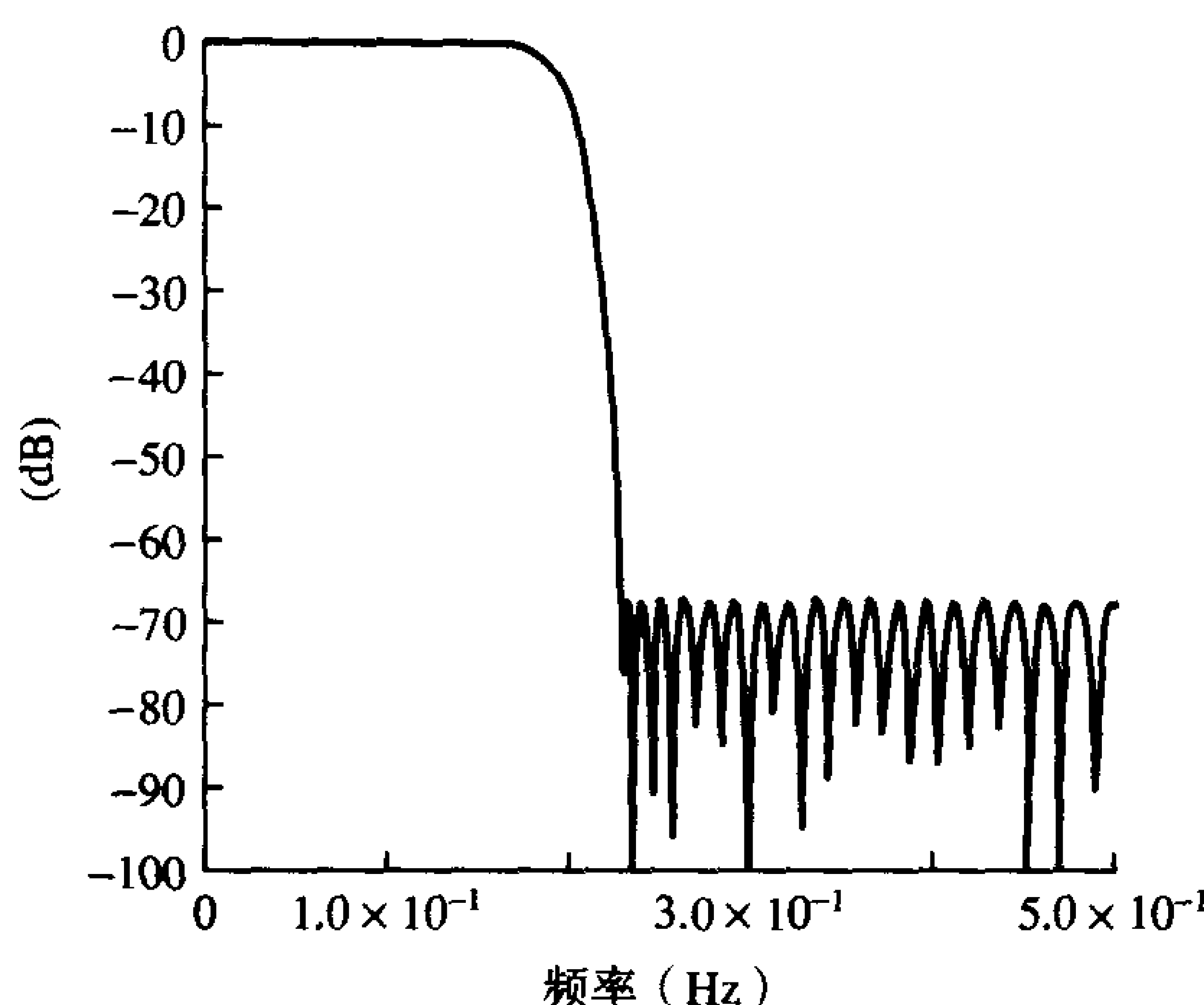


图7.33　例7.19量化前滤波器的频谱图

7.11.2　舍入误差

回想一下给出的FIR滤波器差分方程：

$$y(n)=\sum_{m=0}^{N-1}h(m)x(n-m) \tag{7.46}$$

其中每一个变量都是用固定位数来表示。典型的输入输出样本$x(n-m)$和$y(n)$都是用12位表示，而系数是用16位2的补码格式表示。

从7.46式可以看出，滤波器的输出是由$h(m)$与$x(n-m)$的乘积和得到的。每一个相乘后，积包含的位数要比$h(m)$和$x(n-m)$都多。例如，12位输入乘以16位系数，结果有28位长，在存储到存储器之前需要将该28位的结果量化回16位，或在输出到DAC之前量化回12位。这种量化产生误差，它的影响类似于ADC噪声，但是可能更为严重。量化算术运算结果常用的方法是(a)把结果截断，得到高的有效位而放弃低位；(b)对结果舍入，即选择最接近未舍入结果的高位数据，这是通过给结果加1/2 LSB来实现的。

用两倍长的寄存器准确地表示所有的乘积，然后在得到最终和后对结果进行舍入，即在得到$y(n)$后对$y(n)$做舍入，这样可使舍入误差达到最小。这种方法要比另一种在求和前每个积都做舍入引入的误差小。

7.11.3　溢出误差

溢出误差发生在两个数求和时，通常符号相同的两个大数之和会超出允许的字长。因此，7.46式中，两个积相加时，即$h(0)x(n)$与$h(1)x(n-1)$相加时会发生溢出。

假设最终输出$y(n)$在允许的字长范围内，在部分求和中溢出是微不足道的。这是2的补码算法希望的性质。然而，如果输出$y(n)$超出允许的限制，那么很显然，输出到DAC的样本值将会发生错误，这就需要采取一定措施来避免这种情况的发生。一种方法是检测溢出，然后进行校正，但这种方法花费的代价可能是非常昂贵的。另一种方法是通过对系数进行伸缩（乘一个比例因子）来避免或者允许限制溢出。系数可以用下面两种方法之一进行伸缩：

$$h(m)=\frac{h(m)}{\sum_{k=0}^{N-1}|h(k)|} \tag{7.47a}$$

$$h(m)=\frac{h(m)}{\left[\sum_{k=0}^{N-1}h^2(k)\right]^{1/2}} \tag{7.47b}$$

7.47a 式给定的方法不会出现溢出，但是这种形式的系数伸缩常常是不必要的，因为它是基于溢出的最坏情况，而实际上这种最坏的情况是不可能发生的。而且7.47a式使用的方法要引入比7.47b式给出的方法更多的量化噪声，7.47b 式使用的方法允许偶然溢出。

输入数据的伸缩也可以采用与系数伸缩相类似的方法，通常能得到一个较好的信噪比(SNR)。第三种方法以一种可能达到最好的 SNR 的方法对输入和输出进行伸缩。一种有效的伸缩方法是用一个 2 的幂的伸缩因子。

7.12　FIR 实现技术

FIR 数字滤波器的差分方程为

$$y(n)=\sum_{k=0}^{N-1}h(k)x(n-k) \tag{7.48}$$

系数$h(k)$已在近似阶段求得，选择恰当的结构，进行分析来验证用于表示变量并且执行算术运算的位数。最后阶段就是实现滤波器，这里关键的问题从本质上讲就是产生所选择的滤波器结构的软件代码或硬件实现。这里讨论的是基于 7.48 式表示的横向结构，横向结构是最流行的结构。

考察一下 4.8 式表明，$y(n)$的计算仅包含乘法、加法 / 减法和延迟。因而，为了实现一个滤波器，我们需要下面这些基本的器件：

- 存储器（RAM），存储当前和过去的输入抽样值 $x(n)$和 $x(n-k)$；
- 存储器（RAM 或 ROM），存储滤波器系数 $h(k)$；
- 乘法器（软件或硬件）；
- 加法器或算术逻辑单元（ALU）。

这些器件加上一些控制方法一起组成数字滤波器。如果输入数据源是模拟的，那么我们还需要一个 ADC。类似地，如果输出是模拟的，那么我们需要一个 DAC。因此，实时滤波器的结构具有图 7.34 所描述的形式。按传统的划分，滤波器实现可分为两部分：软件和硬件。然而这种划分在现在的DSP 中显得有点人为化，因为在滤波中用到的大多数器件都是可编程的，几乎没有纯硬件的解决方案。在本书中，我们把那些在大型系统上实现的（例如大型计算机和个人计算机）看作为软件实现。在这些情况下，使用高级语言对滤波方程编程，执行也是离线进行的。使用DSP器件和专用硬件，包括标准的微处理器，我们称为硬件实现。在这些情况下，滤波方程可以按照软硬结合的固件或针对某一器件的汇编语言程序来实现。

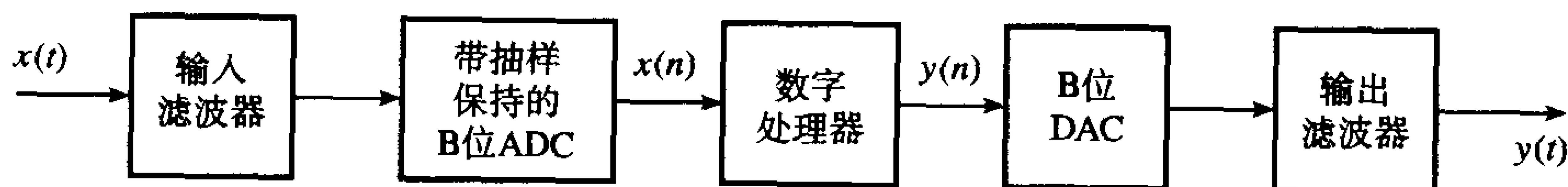


图 7.34　具有模拟输入 / 输出信号的实时数字滤波器的简化框图

在大多数应用中，实时运算常常是主要的目标。在这些情况下，硬件实现是最好的选择。硬件实现能提供最高的速度，但是灵活性差。目前硬件实现常用的有三种方法：标准微处理器（例如摩托罗拉的68000）和DSP处理器（例如德州仪器的TMS320）、构建块及专用算法单元。在构建块方

法中使用了专用硬件块。在专用算法单元和DSP处理器中，滤波所需要的各种器件——乘法器、加法器等是用硬件实现的，并且在单片IC中使用VLSI技术组合在一起。然而，已经构建了专用算法处理器来执行FIR滤波。设计者只需要提供滤波器系数，以及滤波器与外部必要的接口逻辑，例如摩托罗拉的DSP56200和INMOS A100。DSP处理器具有对FIR滤波运算最佳化的结构和指令集。它们要比专用算法处理器更为灵活，但速度要慢一些。

第12章和第13章介绍了利用软件或硬件方法进行系统设计。

图7.35描述了一般的FIR滤波器的运算流程，从这个流程图可看出在每个抽样时刻，我们必须首先将数据移到一个地方，读出并保存最新的输入抽样值$x(n)$，用差分方程计算当前的输出抽样值。

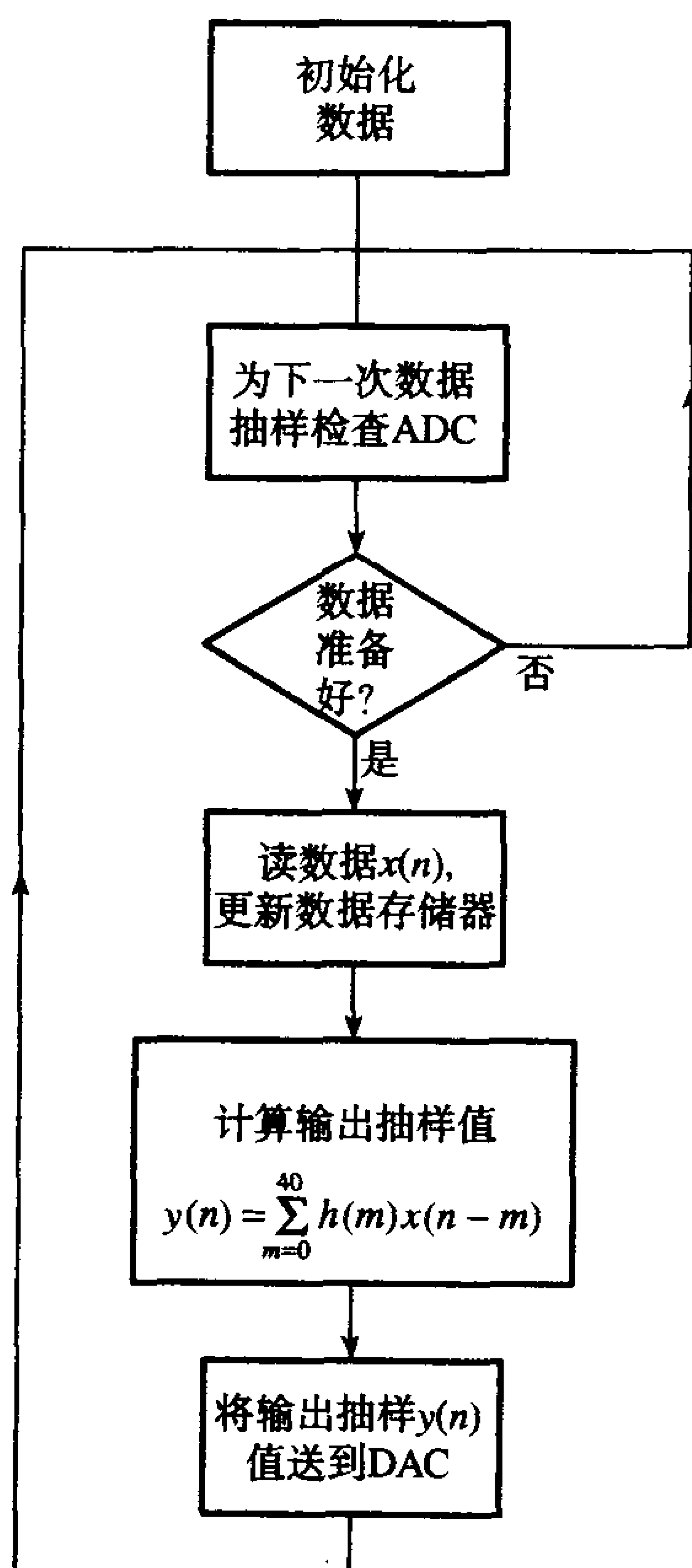


图7.35 实时横向FIR滤波器的简化流程图

7.13 设计实例

例7.20 设计并实现一个线性相位带通滤波器，该滤波器满足以下规范：

通带	900 ~ 1100 Hz
通带波纹	0.87 dB
阻带衰减	>30 dB
抽样频率	15 kHz
系数个数	41

使用TMS32010目标板（参见第13章）实现滤波器。

解：

如前面讨论的那样，FIR 滤波器设计中包括下面 5 个步骤。

- **步骤 1：规范**　规范已经给出。
- **步骤 2：系数计算**　我们使用最佳方法计算滤波器系数，因为最佳方法计算出的滤波器系数（对于非递归 FIR）的个数是最少的，并且最佳方法是实用的。在前面的例子中我们已经计算出了这个滤波器的系数，参见表 7.7。图 7.13 给出了相应的频率响应。
- **步骤 3：实现结构**　选用横向结构（参见图 7.29(a)），因为它可以得出使用 TMS32010 处理器的最有效的实现。这种结构的差分方程为

$$y(n) = \sum_{m=0}^{40} h(m)x(n-m)$$

- **步骤 4：量化和误差分析**　因为使用 TMS32010，所以为了有效地运算，每个系数都应量化为 16 位。为此我们将每一个系数都乘以 2^{15}，然后将结果舍入到最接近的整数。例如，前两个系数量化如下：

$$h(0) = -0.015\,346\,38 \times 2^{15} = -502.87 \approx -503$$

$$h(1) = -0.000\,057\,805\,5 \times 2^{15} = -1.89 \approx -2$$

表 7.16 列出了量化系数和未量化系数。应当检查量化后滤波器的频率响应，验证规范仍然满足，尤其是阻带衰减。量化为 16 位后，我们发现量化后的滤波器的响应和未量化的滤波器的响应的差别很小。

表 7.16　该设计实例的未量化系数 $h(m)$ 和量化系数 $h_q(m)$

m	未量化系数 $h(m)$	量化后的系数 $h_q(m)$
0	−1.534638e−02	−503
1	−5.780550e−05	−2
2	5.023483e−03	165
3	1.266706e−02	415
4	2.108206e−02	691
5	2.776418e−02	910
6	3.005362e−02	965
7	2.586935e−02	848
8	1.444566e−02	473
9	−3.189323e−03	−105
10	−2.416137e−02	−792
11	−4.420712e−02	−1449
12	−5.857453e−02	−1919
13	−6.318557e−02	−2070
14	−5.575461e−02	−1827
15	−3.654699e−02	−1198
16	−8.540099e−03	−280
17	2.308386e−02	756
18	5.201380e−02	1704
19	7.224807e−02	2387
20	7.951681e−02	2606
21	7.224807e−02	2367
22	5.201380e−02	1704
23	2.308386e−02	756
24	−8.540099e−03	−280
25	−3.654699e−02	−1198

（续表）

m	未量化系数 $h(m)$	量化后的系数 $h_q(m)$
26	−5.575461e−02	−1827
27	−6.318557e−02	−2070
28	−5.857453e−02	−1919
29	−4.420712e−02	−1449
30	−2.416137e−02	−792
31	−3.189323e−03	−105
32	1.444566e−02	473
33	2.586935e−02	848
34	3.005362e−02	985
35	2.776418e−02	910
36	2.108206e−02	691
37	1.266706e−02	415
38	5.023482e−03	165
39	−5.780550e−05	−2
40	−1.534638e−02	−503

通过TMS32010处理器，差分方程中部分和的计算将在32位累加器中执行。其中使用了一个位数很长的乘积寄存器（32位）。因此，$N=41$ 时舍入误差的影响很小。在本例中，溢出可忽略掉。即使不能忽略，我们也可以通过将第2步得到的系数除以一个恰当的比例因子SF来克服，例如

$$\mathrm{SF} = \sum_{m=0}^{40} |h(m)|$$

目标板只有一个8位的ADC。这将限制被处理信号的动态范围大约为48 dB。例如，在一个高质量的音频系统中，量化噪声电平是不能接受的，在这样的情况下，ADC的分辨率必须增加。

- **步骤5：实现** 图7.35中给出了FIR滤波运算的流程图。下一步要做的是将流程图转化为TMS32010汇编代码并保存在程序存储器中（参见第12章FIR滤波运算的开发及编程）。

7.14 小结

数字滤波器的设计可分为5个独立的阶段：滤波器技术规范、系数计算、实现结构、误差分析和滤波器实现。

滤波器技术规范与应用有关，且应该包括振幅和相位特性的规范。

系数计算本质上就是求出满足所期望的规范的$h(m)$值。计算FIR滤波器系数最常用的方法有三种：(1) 窗口方法，(2) 频率抽样方法，(3) 最佳方法。窗口方法是最容易的，但是缺乏灵活性，特别是当通带波纹和阻带波纹不同时更是如此。频率抽样方法非常适合FIR滤波器的递归实现，频率抽样法也适合那些除了要求标准频率选择性滤波器（低通、高通、带通和带阻）之外的滤波器。最佳方法是最高效和灵活的一种设计方法。上述三种方法，本章都已做了详细的阐述。

三种最常用的FIR滤波器结构是横向结构、频率抽样结构和快速卷积结构。横向结构包含一个使用滤波器系数的直接卷积；频率抽样结构直接同系数计算的频率抽样方法相联系。结构的选择与具体的应用有关。

长字长的或者高阻带衰减的FIR滤波器的性能可能会受到有限字长的影响。例如，系数量化后它们的频率响应可能会发生变化。因而应当对这些滤波器的特性进行检查以确保允许的合适的字长，特别是当字长小于12时。

在完成好前四步后，通常要考虑实现问题，以及考虑软件编程或选择结构的硬件实现。

7.15　FIR 滤波器的应用实例

FIR 滤波器在许多领域都已得到应用，如多速率处理（Crochiere and Rabiner, 1981）、噪声抑制（Hamer et al., 1985）、匹配滤波（参见第 13 章）和图像处理（Wade et al., 1990）。

例如在多速率处理中，FIR滤波器已经成功地应用于多速率系统的有效的数字抗混叠和抗像频滤波中。这些多速率系统包括高质量的数据采集和激光唱盘播放器（参见第 9 章）。

习题

FIR 滤波器的概念

7.1　类型 2 的线性相位 FIR 滤波器的频率响应 $H(\omega)$表示为（参见表 7.1）

$$H(\omega) = \mathrm{e}^{-j\omega(N-1)/2} \sum_{n=1}^{N/2} b(n) \cos[\omega(n - \tfrac{1}{2})]$$

其中 $b(n)$与滤波器系数有关。请解释为什么具有上述频率响应的滤波器不适合作为高通滤波器。并用简单的情况（如 $N = 4$）来说明你的答案。

7.2　一个 FIR 滤波器的冲激响应 $h(n)$定义在区间 $0 \leqslant n \leqslant N-1$ 上，证明：如果 N是偶数，$h(n)$满足正对称条件，即 $h(n) = h(N-n-1)$，那么滤波器有一个线性相位响应，求滤波器的幅度和相位响应的表达式。

窗口方法

7.3　证明理想带通滤波器（参见表 7.2）的冲激响应为

$$\begin{aligned} h_D(n) &= 2f_2 \frac{\sin n\omega_2}{n\omega_2} - 2f_1 \frac{\sin n\omega_1}{n\omega_1} & n \neq 0 \\ &= 2(f_2 - f_1) & n = 0 \end{aligned}$$

这里 f_1 是下通带频率，f_2 是上通带频率。

7.4　(1) 用窗口方法求满足下列规范的 FIR 低通数字滤波器的系数：

阻带衰减	50 dB
通带边沿频率	3.4 kHz
过渡带宽	0.6 kHz
抽样频率	8 kHz

在你的答案中应包括使用的窗口类型以及选择的理由。

(2) 假设滤波器系数存储在微型计算机的连续的存储单元内，试按系数的存储顺序列出系数值。

(3) 画出并简单地描述实时滤波器的直接软件实现的流程图，并提出两种改善软件实现效率的方法。

提示：在设计中可以利用表 7.2 中给出的信息。

最佳（Parks-McClellan）方法

7.5　(1) 设有一个冲激响应满足下列对称条件的线性相位 FIR 滤波器：

$$h(n) = h(N-n-1),$$
$$n = 0, 1, \ldots, (N-1)/2$$

式中N为滤波器系数个数。假设N为偶数，求滤波器的幅度和相位响应，并且证明滤波器具有恒定的相位和群延迟。试解释数字滤波器中线性相位响应的实际意义。

(2) 在某个信号分析仪中，为了提取特征信息需要线性相位带通数字滤波器。该滤波器要求满足以下规范：

通带	12 ~ 16 kHz
过渡带宽	3 kHz
抽样频率	96 kHz
通带波纹	0.01 dB
阻带衰减	80 dB

假设使用最佳方法（Remez交换法）计算滤波器系数，确定滤波器的下列这些参数：

(a) 滤波器系数的个数N；

(b) 合适的滤波器频带的权值；

(c) 带沿频率，以适合最佳方法的形式表示。

简单地解释权值和网格频率在最佳方法中扮演的角色。

试为上述问题提出一个合适的网格密度，可以使用表7.17给出的信息。

表7.17　对于带通滤波器，估计长度N的关系

$$N \approx \frac{C_\infty(\delta_p, \delta_s)}{\Delta F} + g(\delta_p, \delta_s)\Delta F + 1$$

其中

$$C_\infty(\delta_p, \delta_s) = [\log_{10}\delta_s][b_1(\log_{10}\delta_p)^2 + b_2\log_{10}\delta_p + b_3] + [b_4(\log_{10}\delta_p)^2 + b_5\log_{10}\delta_p + b_6]$$

$$g(\delta_p, \delta_s) = -14.6\log_{10}\left(\frac{\delta_p}{\delta_s}\right) - 16.9$$

$b_1 = 0.012\,02$　　$b_2 = 0.096\,64$

$b_3 = -0.513\,25$　　$b_4 = 0.002\,03$

$b_5 = -0.570\,5$　　$b_6 = -0.443\,14$

ΔF，是用抽样频率归一化的过渡带宽

δ_p，通带波纹或偏差

δ_s，阻带波纹或偏差

7.6　一个FIR低通数字滤波器需满足下面的参数：

阻带衰减	> 40 dB
通带带沿频率	100 Hz
通带波纹	< 0.05 dB
过渡带宽	10 Hz
抽样频率	1024 Hz

(1) 计算并列出滤波器系数，并说明使用的方法及选择该方法的理由。

(2) 使用快速卷积方法实现滤波器的实时运算。概括说明你是怎样通过重叠保留技术利用FT实时实现滤波器的。清楚地指出一些参数，如输入部分重叠的抽样数、这些部分的长度、用来变换的大小及输出抽样值是如何从变换中提取的。

7.7　设计一个线性相位 41 点 FIR 差分器，差分器应该满足下列规范：

通带边沿频率	1 kHz
阻带边沿频率	1.5 kHz
抽样频率	10 kHz
通带偏差	0.01
阻带偏差	0.01

使用最佳方法（Parks-McClellan/Remez 交换算法）计算差分器的系数，并画出它的幅度－频率响应。

7.8　设计一个线性相位 43 点 FIR 希尔伯特变换滤波器，该滤波器应该满足下列技术规范：

下带沿频率	1 kHz
上带沿频率	4.5 kHz
抽样频率	10 kHz
通带偏差	0.01

使用最佳方法计算该滤波器的系数，并画出其以 dB 为单位的幅度－频率响应。

频率抽样滤波器

7.9　4 点线性相位 FIR 滤波器由下面的频率抽样值刻画：

$$
|H(k)| = \begin{cases} 1, & k=0 \\ \frac{1}{2}, & k=1,3 \\ 0, & k=2 \end{cases}
$$

(a) 从 7.24 式给出的传递函数的一般表达式出发，证明上面滤波器的传递函数含有 4 个零点和 3 个极点。

(b) 画出该滤波器的极零图。

(c) 画出该滤波器的频率响应。

(d) 使用频率抽样结构，开发并画出带有复共轭极点组合的滤波器实现框图。

(e) 求滤波器的 4 个系数，这些系数必须是实的。

7.10　4 点线性相位频率抽样 FIR 滤波器由下列频率抽样值来刻画：

$$
|H(k)| = \begin{cases} 1, & k=0 \\ 0, & k=1,2,3 \end{cases}
$$

(a) 从 7.24 式给出的传递函数的一般表达式出发，求滤波器传递函数的零点数和极点数。

(b) 画出该滤波器的极零图。

(c) 使用频率抽样结构，开发并画出带有复共轭极点组合的滤波器实现结构。

(d) 求滤波器的 4 个系数，这些系数必须是实的。

7.11　频率抽样滤波器既与 FIR 滤波器具有某些相同的特性又与 IIR 滤波器具有另外一些相同的特性。在该习题中我们将考虑特性中的一些问题：

(a) 递归频率抽样滤波器相对于非递归频率抽样滤波器有哪些主要的优点？

(b) 讨论与递归频率抽样滤波器有关的有限字长效应问题，在实际应用中如何克服？

(c) 图 7.36 中描述了带通频率抽样滤波器的极零图。

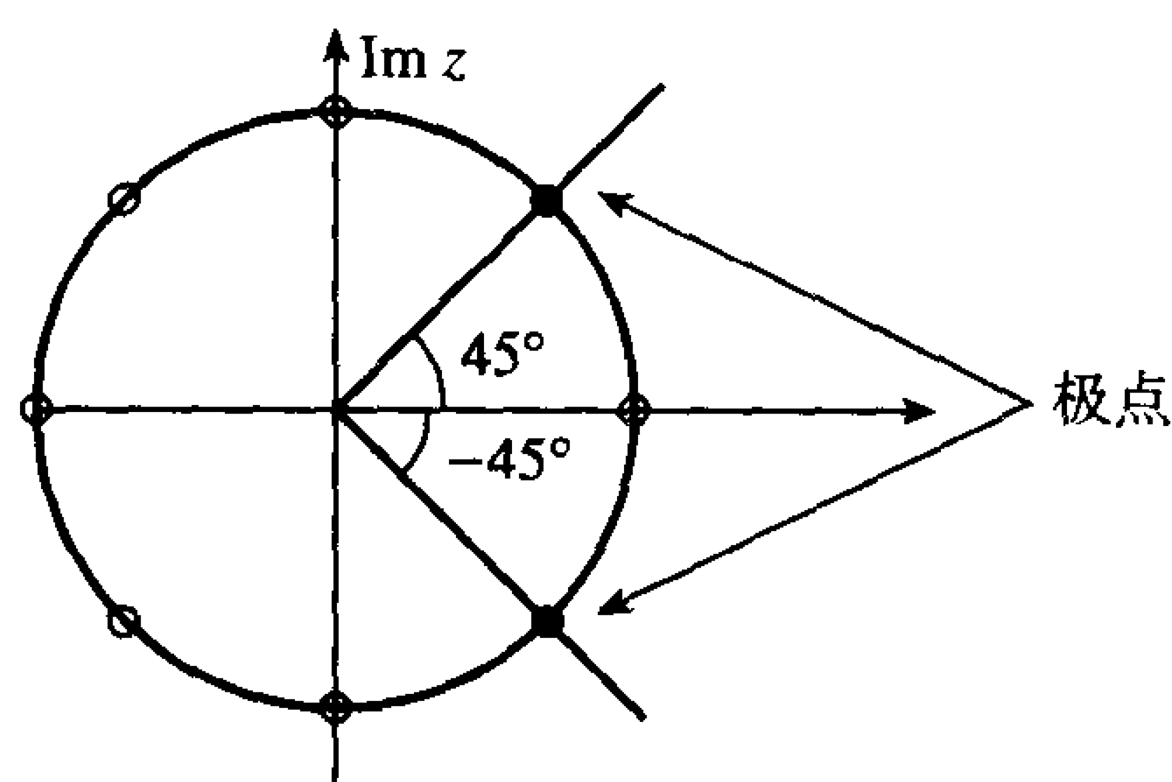
(i) 由极零图写出带通滤波器在如下频率处的频率抽样值 $H(k)$，

$$\omega_k = \frac{2\pi k}{N}, \quad k = 0, 1, \ldots, 7$$

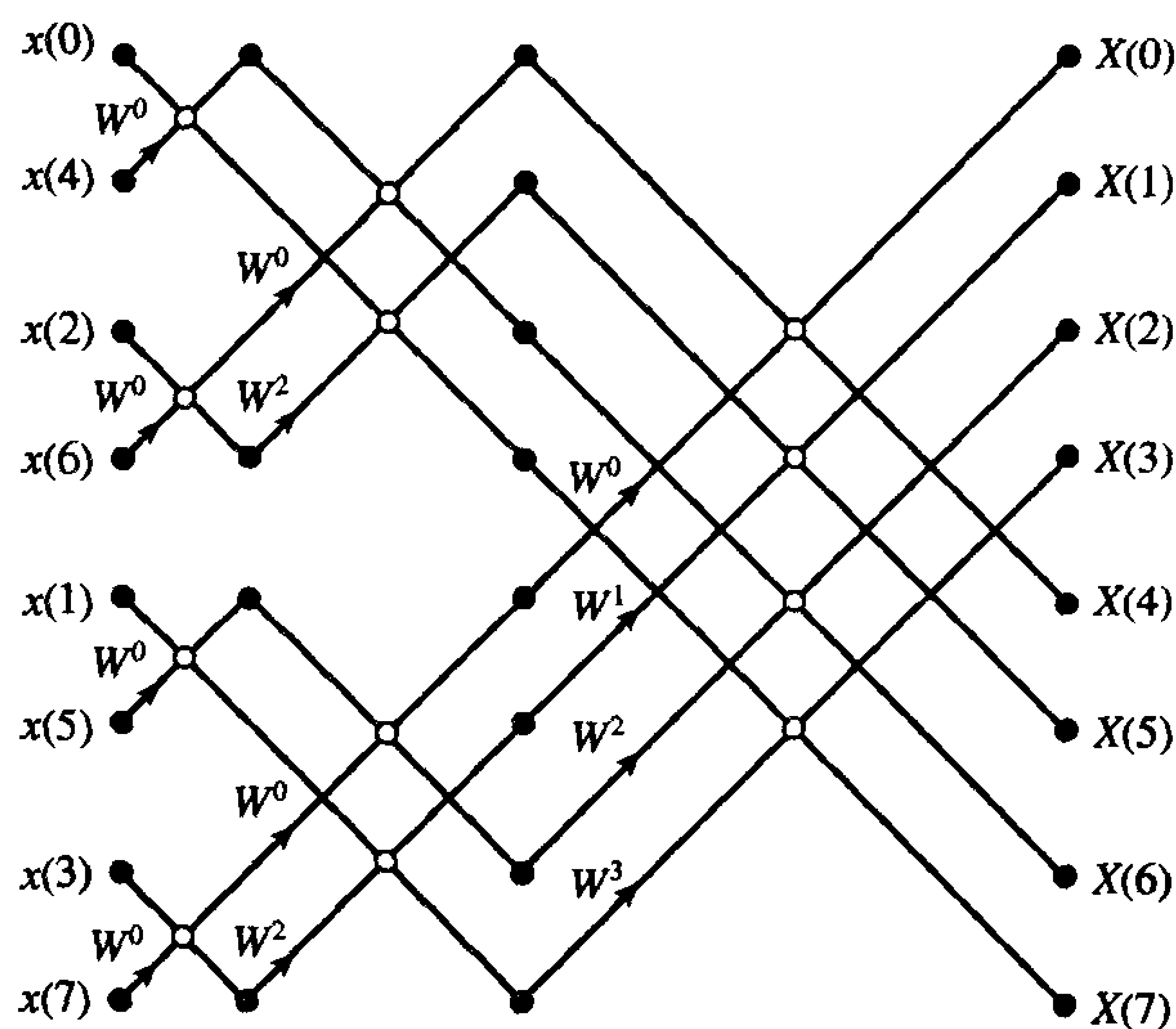
说明其中做出的合理假设。

(ii) 画出滤波器的幅度 - 频率响应，清楚地标注出抽样时刻。

(iii) 计算滤波器的系数，即求它的递归形式的传递函数 $H(z)$。



(a)



NB $W^0 = 1, W^1 = \frac{\sqrt{2}}{2} - j\frac{\sqrt{2}}{2}, W^2 = -j, W^3 = -\frac{\sqrt{2}}{2} - j\frac{\sqrt{2}}{2}$

(b)

图 7.36 带通频率抽样滤波器的极零图和基 -2 FFT 流图

(d) 解释从上面得到的频率抽样值是如何由图 7.36(b)的基 -2 FFT 流图计算滤波器的冲激响应。

7.12 (a) 借助框图解释频率抽样滤波器设计方法中的基本概念。

(b) 满足下列规范的低通数字滤波器存在要求：

通带	0 ~ 20 Hz
抽样频率	300 Hz
阻带衰减	> 50 dB
滤波器长度	15

(c) 由表7.18中的信息，使用频率抽样方法，求递归形式的数字滤波器的传递函数的系数。

(i) 开发并画出滤波器的实现结构图，并比较递归实现和直接形式 FIR 实现的存储量和计算量。

(ii) 解释说明为什么上面传递函数表示的滤波器仍是 FIR 滤波器？尽管它的传递函数表明它是一个递归滤波器。解释在实际应用中递归频率抽样滤波器可能会碰到哪些困难，并说明如何克服这些困难。

表 7.18 类型 1 低通频率抽样滤波器在 $N = 15$ 时最佳过渡带频率抽样值（取自 Rabiner et al., 1970）

BW	阻带衰减（dB）	T_1	T_2	T_3
一个过渡带频率抽样值, $N = 15$				
1	42.309 322 83	0.433 782 96		
2	41.262 992 86	0.417 938 23		
3	41.253 337 86	0.410 473 63		
4	41.949 077 13	0.404 058 84		
5	44.371 245 38	0.392 681 89		
6	56.014 165 88	0.357 665 25		
二个过渡带频率抽样值, $N = 15$				
1	70.605 405 85	0.095 001 22	0.589 954 18	
2	69.261 681 56	0.103 198 24	0.593 571 18	
3	69.919 734 95	0.100 836 18	0.589 432 70	
4	75.511 722 56	0.084 074 93	0.557 153 12	
5	103.460 783 00	0.051 802 06	0.499 174 24	
三个过渡带频率抽样值, $N = 15$				
1	94.611 661 91	0.014 550 78	0.184 578 82	0.668 976 13
2	104.998 130 80	0.010 009 77	0.173 607 13	0.659 515 26
3	114.907 193 18	0.008 734 13	0.163 973 10	0.647 112 64
4	157.292 575 84	0.003 787 99	0.123 939 63	0.601 811 54

BW 称为通带内的频率抽样数。

7.13 图 7.37 给出了一个简单的频率抽样带通滤波器的极零图。

(a) 画出滤波器的幅度 - 频率响应，并写出在抽样点处的幅度 - 频率响应值。

(b) 由 7.24 中频率抽样滤波器的一般传递函数，写出该滤波器的传递函数，并解释你的答案。

(c) 画出滤波器的实现结构图，并写出差分方程。

(d) 根据计算量和存储量，将频率抽样实现与直接方式实现进行比较。

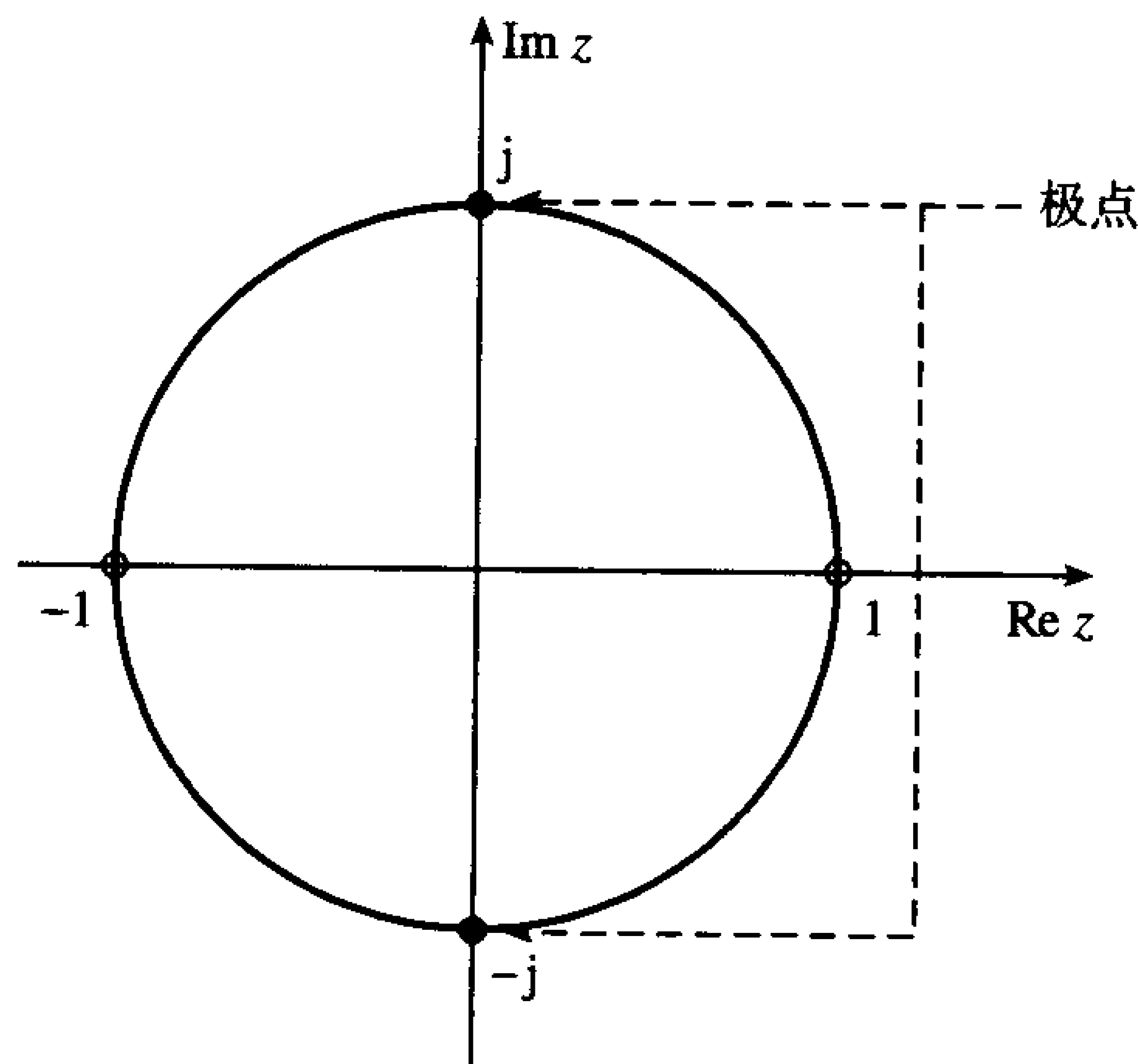


图 7.37 一个简单的频率抽样带通滤波器的极零图

7.14 图7.38给出了一个简单的频率抽样带通滤波器的极零图。

(a) 画出梳状滤波器部分（只包含零点）的幅度－频率响应。

(b) 画出滤波器的幅度－频率响应，并写出在抽样点处的幅度－频率响应值。

(c) 写出滤波器递归形式的传递函数。

(d) 从7.24式频率抽样滤波器的一般传递函数出发，求该滤波器的传递函数，并解释你的答案。

(e) 画出滤波器的实现框图，并写出差分方程。

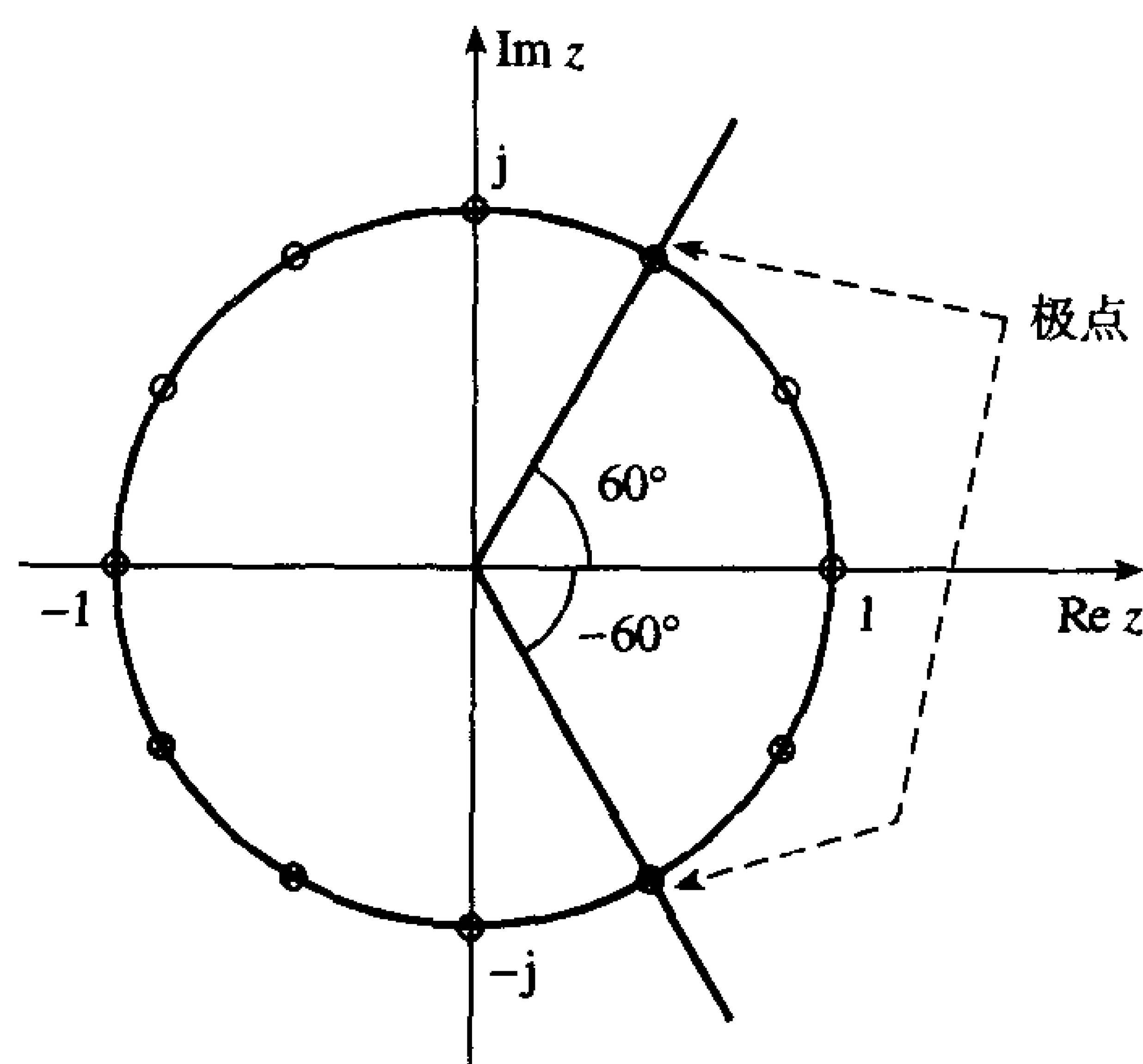


图7.38 一个简单的频率抽样带通滤波器的极零图

7.15 (1) 简单讨论一下具有线性相位特性的可实现的数字滤波器的必要条件，以及具有这一特性的滤波器的优点。

(2) 在某信号处理应用中，有效的输入信号频率范围为$0 \leqslant f \leqslant 10$ Hz，信号受到一个50 Hz的电源干扰的污染。因此决定在以每秒500个抽样值的速率数字化复合信号之后，使用线性相位数字滤波器将干扰消除。设计该滤波器的第一步就是建立图7.39中给出的极零图，求滤波器的传递函数$H(z)$和它的差分方程。

(3) 在(2)中得到的滤波器要在微型计算机用简单的算术运算实现，这些算术运算包括加/减和移位。重新设计滤波器使它的系数为整数，这不会使滤波器系数个数和抽样速率增加。

(4) 证明(3)中给出的滤波器的相位响应表示为

$$\theta(\omega) = -\omega T$$

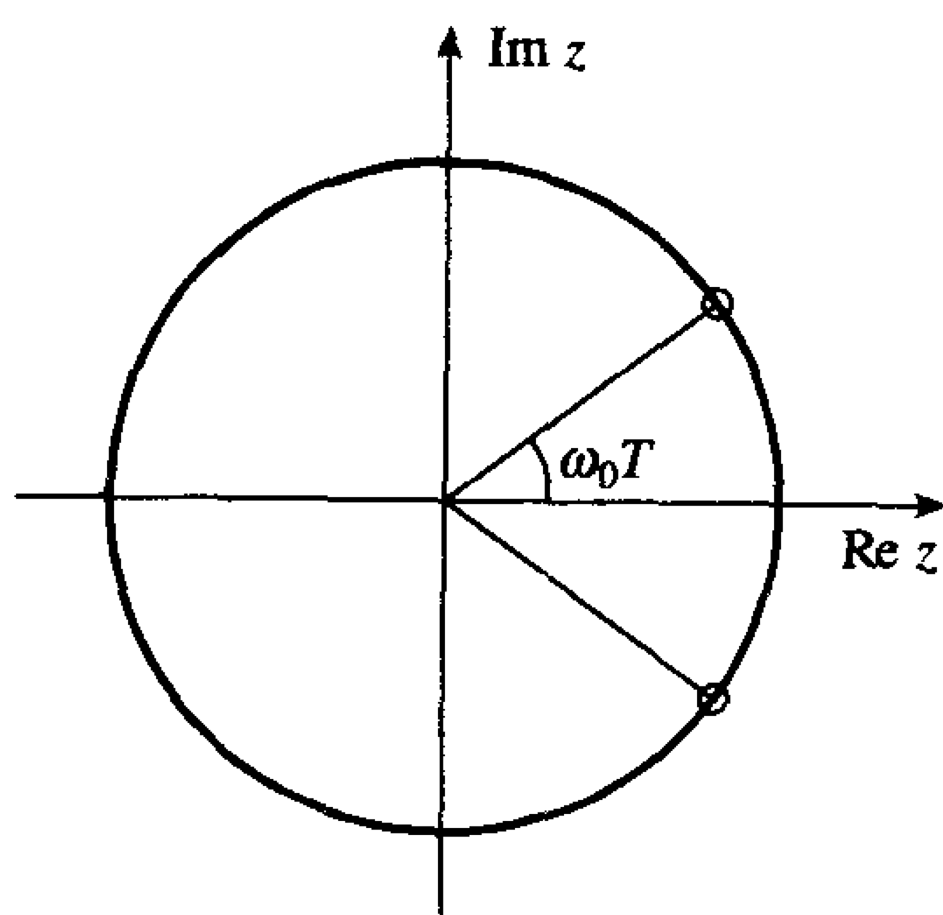


图7.39 习题7.15的极零图，其中$\omega_0 T = \pi/5$弧度

7.16 (1) 生物医学应用系统的实时窄带线性相位数字滤波器存在要求，证明频率抽样滤波器可应用到该系统。假设 N 点频率抽样滤波器的传递函数为

$$H(z) = \frac{1 - r^N z^{-N}}{N}$$

$$\left(\sum_{k=1}^{M} \frac{|H(k)| [2\cos(2\pi k\alpha/N) - 2r\cos[2\pi k(1+\alpha)/N]}{1 - 2r\cos(2\pi k/N) z^{-1} + r^2 z^{-2}} + \frac{H(0)}{1 - z^{-1}} \right)$$

其中 $H(k)$ 是按抽样间隔为 F_s/N（$\alpha = (N-1)/2$）对期望的频率响应进行抽样所得到的抽样值。

(2) 期望的滤波器由下列规范来刻画：

通带	48 ~ 52 Hz
过渡带宽	2 Hz
抽样频率	500 Hz
阻带衰减	> 60 dB

指定合适的频率抽样值 $|H(k)|$，建立并画出滤波器的实现结构框图，将存储量和计算量与等效的横向滤波器进行比较。

(3) 解释上面的 $H(z)$，并且解释递归频率抽样滤波器在实际应用中遇到的困难，指出如何克服这些困难。

解释为什么 $H(z)$ 描述的是一个递归滤波器，而且它的单位冲激响应 $h(n)$ 是有限持续时间的。

7.17 具有下面频率响应的 N 点 FIR 滤波器存在要求：

$$H(e^{j\omega}) = |H(e^{j\omega})| e^{-j\omega\alpha}$$

式中 $\alpha = (N-1)/2$，假定 $H(e^{j\omega})$ 的抽样值是以间隔 $f_k = (k+1/2)\, F_s/N (k = 0, 1, ..., N-1)$ 进行抽样的。

(1) 证明：当 N 为偶数时，冲激响应可表示为

$$h(n) = \frac{1}{N} \left\{ \sum_{k=0}^{N/2-1} 2|H(k)| \cos[2\pi(n-\alpha)(k+1/2)/N] \right\}$$

(2) 证明：当 N 为奇数时，冲激响应可表示为

$$h(n) = \frac{1}{N} \left\{ \sum_{k=0}^{(N-3)/2} 2|H(k)| \cos[2\pi(n-\alpha)(k+1/2)/N] + H[(N-1)/2] \cos[\pi(n-\alpha)] \right\}$$

(3) 分别求出(1)和(2)中传递函数 $H(z)$ 的递归形式表达式。

专用 FIR 滤波器

7.18 高通 FIR 滤波器由下面的冲激响应系数来刻画：

$$\{h(n)\} = \{0.127, -0.026, -0.237, 0.017, 0.434\}$$

借助 7.9.2 节给出的频率变换写出等效的低通滤波器的系数。

7.19 计算用凯塞窗函数的 FIR 半频带滤波器的系数。半频带滤波器应该满足下面这些参数：

通带波纹	0.5 dB
阻带衰减	45 dB

通带带沿频率	2 kHz
抽样频率	10 kHz

7.20 使用最佳方法重复习题7.19。

FIR 滤波器的实现

7.21 一个模拟信号被一个50 Hz的分量以及它的谐波100 Hz、150 Hz、200 Hz、250 Hz和300 Hz所污染。假设被污染信号以1 kHz被抽样和被数字化。

求消除干扰和它的谐波的简单数字滤波器的传递函数。画出数字滤波器的实现结构框图。比较并对照有限字长对数字滤波器性能的影响以及容错元件对模拟滤波器性能的影响。使用陷波滤波器来说明你的答案。

7.22 (1) 评估有限字长约束对实时FIR数字滤波器实现的影响，如何使这种影响最小？

(2) 在某实时数字信号处理系统中，N点FIR滤波器的每个系数被表示为n位2的补码数。证明最大阻带衰减A_{max}的上界可表示为

$$A_{max} < 20\log_{10} N2^{-B}$$

式中B是系数的字长，N为滤波器长度。

说明其中做出的任何假设，并且解释以上给出的边界。

(3) 下面列出的是7点FIR滤波器的系数，画出滤波器的实现结构框图，滤波器应使每个输出计算的乘法次数最少。

$$h(0) = -0.3$$
$$h(1) = 0.4$$
$$h(2) = 0.2$$
$$h(3) = 0.5$$
$$h(4) = 0.2$$
$$h(5) = 0.4$$
$$h(6) = -0.3$$

7.23 用2的补码分数算术实现定点FIR数字滤波器，系数用3位（包括符号位）表示。

(1) 计算并列出所有能够表示的小数。并指出能表示的最大和最小的小数。

(2) 下面是FIR滤波器未量化的系数。假定系数截断后被量化为3位（包括符号位）。列出量化后的系数及系数的量化误差。

n	$h(n)$
0	−0.149 75
1	0.256 872
2	0.699 40
3	0.256 872
4	−0.149 725

(3) 如果系数被舍入，重复(2)。

7.24 一个FIR滤波器系数为$\{h(n)\} = \{-1, 0.5, 0.75\}$。

(1) 画出滤波器横向实现的结构图。

(2) 假设系数和输入数据抽样值在截断后用3位表示（包括符号位）。求量化后的系数值，用二进制和十进制列出系数表。

(3) 证明：如果数据$\{x(n)\} = \{0.5, -1, -0.5\}$加到滤波器，其输出$y(n)$仍是正确的，尽管中间结果溢出（设用的是双字长累加器）。

(4) 证明：当输入数据$\{x(n)\} = \{-1, -0.75, 0.5\}$时，由于溢出会得出错误的输出结果。如何预防溢出？

7.25 设计一个减少生理噪声的实时低通滤波器，滤波器将是一个较大的数字信号处理系统的一部分，因此滤波器中的运算应该尽可能少。

该滤波器应满足下面这些幅度规范：

通带	8 ~ 12Hz
通带波纹	0.1 dB
过渡带宽	2 Hz
阻带衰减	30 dB
抽样频率	100 Hz

其他要求是

(1) 希望带内信号分量之间的谐波关系失真最小；

(2) 将模拟输入数字化成 12 位，滤波器用 TMS320C25 DSP 处理器实现。

7.26 设计一个多频带 FIR 数字滤波器，要求满足下列规范：

频带 1	0 ~ 0.5 kHz	
	阻带衰减	49 dB
频带 2	1 ~ 1.5 kHz	
	通带波纹	0.3 dB
频带 3	1.8 ~ 2.5 kHz	
	阻带衰减	38 dB
频带 4	3 ~ 3.6 kHz	
	通带波纹	0.3 dB
频带 5	4.1 ~ 5 kHz	
	阻带衰减	55 dB

使用由 TMS320C25 处理器、抽样频率为 12 位 ADC 和 12 位 DAC 的构成的系统来实现滤波器。

7.27 讨论数字滤波器设计的五个主要步骤，用下面的设计问题说明你的答案。

数字滤波器是用于实时生理噪声抑制。滤波器应该满足下列规范：

通带	0 ~ 10 Hz
阻带	20 ~ 64 Hz
抽样频率	256 Hz
最大通带波纹	0.026 dB
阻带衰减	30 dB

其他一些重要的要求是

(1) 滤波器应该具有线性相位特性，以便在信号频带内尽可能小地引入失真。

(2) 滤波的有效时间是有限的，滤波是一个更大的处理过程中的一部分。

(3) 滤波器使用 TMS32010 处理器来实现，输入被数字化为 12 位。

MATLAB 习题

7.28 使用 MATLAB 计算系数，画出幅度 - 频率响应，用 dB 表示，并且确定下面每一个基于窗口方法的滤波器的零点位置（假设滤波器抽样频率是 2 kHz，并且使用哈明窗函数）：

(1) 7点带通FIR滤波器，通带带沿频率和阻带带沿频率分别是200 Hz和500 Hz。

(2) 8点带通FIR滤波器，通带沿频率和阻带沿频率分别是200 Hz和500 Hz。

(3) 7点FIR差分器，通带带沿频率和阻带沿频率分别是200 Hz和500 Hz。

(4) 8点FIR希尔伯特变换器，带沿频率为200 Hz和500 Hz。

解释上述滤波器在零点位置上的异同之处。

7.29 使用窗口方法设计41点带通FIR滤波器来近似下面的理想幅度响应特性：

$$H(f) = \begin{cases} 1 & 2\ \text{kHz} \leqslant f \leqslant 4\ \text{kHz} \\ 0 & \text{其他} \end{cases}$$

确定滤波器的冲激响应系数，并使用MATLAB方法，用于下面两种情况的幅度和相位频率响应：

(1) 使用矩形窗；

(2) 使用哈明窗。

7.30 一个满足下列规范的线性相位低通最佳FIR滤波器存在要求：

滤波器长度	21
通带带沿频率	2 kHz
阻带带沿频率	3 kHz
抽样频率	10 kHz

(1) 借助MATLAB计算滤波器的系数并画出它的幅值响应（用dB表示）和相位响应（用度表示）。

(2) 计算并画出滤波器的相位和群延迟响应。

(3) 通过考察幅度和相位响应，确定零点的位置。

(4) 解释为什么相位响应不连续，如何校正相位响应中的跳跃点。

7.31 设计一个FIR数字滤波器，要求满足下列规范：

通带	150 ~ 250 Hz
过渡带宽	50 Hz
通带波纹	0.1 dB
阻带衰减	60 dB
抽样频率	1 kHz

使用哈明窗和MATLAB计算滤波器系数。

7.32 要求一个线性相位FIR带通滤波器满足下列规范：

通带	8 ~ 12 kHz
阻带波纹	0.001
通带波纹	0.01
抽样频率	48 kHz
过渡带宽	3 kHz

借助MATLAB求下列情况的幅度－频率响应：

(1) 使用哈明窗；

(2) 使用凯塞窗；

(3) 使用最佳方法；
(4) 使用频率抽样方法。
比较这四种情况。

7.33 在某信号分析仪提取特征信息时需要一个线性相位带通数字滤波器。滤波器要求满足下列规范：

通带	12 ~ 16 kHz
过渡带宽	3 kHz
抽样频率	96 kHz
通带波纹	0.01 dB
阻带衰减	80 dB

使用最佳方法计算滤波器系数。借助 MATLAB 确定：
(1) 滤波器系数个数 N；
(2) 滤波器的系数。
画出滤波器幅度 – 频率响应图。

7.34 设计一个多波带 FIR 数字滤波器，要求满足下列规范：

频带 1：0 ~ 0.5 kHz，阻带衰减≥ 49 dB
频带 2：1 ~ 1.5 kHz，通带波纹，0.3 dB
频带 3：1.8 ~ 2.5 kHz，阻带衰减，38 dB
频带 4：3 ~ 3.6 kHz，通带波纹，0.3 dB
频带 5：4.1 ~ 5 kHz，阻带衰减，55 dB

使用最佳方法和 MATLAB 计算滤波器系数，并画出幅度 – 频率响应。假设抽样频率为 10 kHz，过渡带宽为 100 Hz。

7.35 设计一个最佳低通滤波器，要求满足下列规范：

通带	0 ~ 6 kHz
过渡带宽	1 kHz
通带波纹	0.1 dB
阻带衰减	50 dB
抽样频率	16 kHz

借助 MATLAB 命令 remezord 和 remez，求滤 LH 器长度和系数。画出滤波器的幅度 – 频率响应。

7.36 使用凯塞窗函数和 MATLAB 计算一个 FIR 半波段滤波器的系数。半波段滤波器满足下面这些参数：

通带波纹	0.5 dB
阻带衰减	45 dB
通带沿频率	2 kHz
抽样频率	10 kHz

使用最佳方法和 MATLAB 命令 remezord 和 remez 重新回答习题 7.19。

7.37 设计一个 41 点的线性相位 FIR 差分器，要求满足下列规范：

通带带沿频率	1 kHz
阻带带沿频率	1.5 kHz
抽样频率	10 Hz
通带偏差	0.01
阻带偏差	0.01

使用最佳方法（Parks-McClellan算法）和MATLAB计算差分器的系数。画出该差分器的幅度－频率响应。

7.38 设计一个 43 点的线性相位 FIR 希尔伯特变换滤波器，要求满足下列规范：

下带沿频率	1 kHz
上带沿频率	4.5 kHz
抽样频率	10 kHz
通带偏差	0.01

使用最佳方法（Parks-McClellan算法）和MATLAB计算希尔伯特变换器的系数。画出它的幅度－频率响应曲线，用 dB 表示。

参考文献

Crochiere R.E. and Rabiner L.R. (1981) Interpolation and decimation of digital signals – a tutorial review. *Proc. IEEE*, **69**(3), 300–31.

Hamer C.F., Ifeachor E.C. and Jervis B.W. (1985) Digital filtering of physiological signals with minimal distortion. *Medical and Biol. Eng. and Computing*, **23**, 274–8.

Harris S.P. and Ifeachor E.C. (1998) Automatic design of frequency sampling filters by hybrid Genetic Algorithm Techniques. *IEEE Transactions on Signal Processing*, **46**(12), December, 3304–14.

Herrman O., Rabiner R.L. and Chan D.S.K. (1973) Practical design rules for optimum finite impulse response digital filters. *Bell System Technical J.*, **52**, 769–99.

Ifeachor E.C. and Harris S.P. (1993) A new approach to frequency sampling filter design, in *Proc. IEE/IEEE Workshop Natural Algorithms in Signal Processing*, 5/1–8.

Lawrence V.B. and Salazar A.C. (1980) Finite precision design of linear-phase FIR filters. *Bell System Technical J.*, **59**(9), 1575–98.

Lynn P.A. (1973) Recursive digital filters with linear phase characteristics. *Computer J.*, **15**, 337.

Lynn P.A. (1975) Frequency sampling filters with integer multipliers. In *Introduction to Digital Filtering*, Bogner R.E. and Constantinides A.G. (eds). New York: Wiley.

McClellan J.H., Parks T.W. and Rabiner L.R. (1973) A computer program for designing optimum FIR linear phase digital filters. *IEEE Trans. Audio Electroacoustics*, **21**, 506–26.

Mintzer F. and Liu B. (1979) Practical design rules for optimum FIR bandpass digital filters. *IEEE Trans. Acoustics, Speech Signal Processing*, **27**(2), 204–6.

Mitra S.K. and Kaiser J.F. (1993) *Handbook for Digital Signal Processing*. New York: Wiley.

Oppenheim A.V. and Schaffer R.W. (1975) *Digital Signal Processing*. Englewood Cliffs NJ: Prentice-Hall.

Parks T.W. and Burrus C.S. (1987) *Digital Filter Design*. New York: Wiley.

Rabiner L.R. and Gold B. (1975) *Theory and Applications of Digital Signal Processing*. Englewood Cliffs NJ: Prentice-Hall.

Rabiner L.R., Gold B. and McGonegal C.A. (1970) An approach to the approximation problem for nonrecursive digital filters. *IEEE Trans. Audio Electroacoustics*, **18**, 83–106.

Suckley D. (1990) Genetic algorithm in the design of FIR filters. *IEE Proc. Part G*, **138**(2), 234–8.

Wade G., van Eetvelt P. and Darwen H. (1990) Synthesis of efficient low-order FIR filters from primitive sections. *IEE Proc. Part G*, **137**(5), 367–72.

参考书目

Bateman A. and Yates W. (1988) *Digital Signal Processing Design.* London: Pitman.

Chan D.S.K. and Rabiner L.R. (1973) Analysis of quantization errors in the direct form for finite impulse response digital filters. *IEEE Trans. Audio Electroacoustics*, **21**(4), 354–66.

Chan D.S.K. and Rabiner L.R. (1973) An algorithm for minimizing roundoff noise in cascade realizations of finite impulse response digital filters. *Bell System Technical J.*, **52**(3), 347–85.

DeFatta D.J., Lucas J.G. and Hodgkiss W.S. (1988) *Digital Signal Processing: A System Design Approach.* New York: Wiley.

Gersho A., Gopinath B. and Odlyzko A.M. (1979) Coefficient inaccuracy in transversal filtering. *Bell System Technical J.*, **58**(10), 2401–2416.

Gold B. and Jordan K.L., Jr (1968) A note on digital filter synthesis. *Proc. IEEE (Lett.)*, **56**, 1717–18.

Gold B. and Jordan K.L., Jr (1969) A direct search procedure for designing finite duration impulse response filters. *IEEE Trans. Audio Electroacoustics*, **17**, 33–6.

Gold B. and Rader C.M. (1969) *Digital Processing of Signals.* New York: McGraw-Hill.

Gore A.E. (1986) Cascadable digital signal processor. *New Electronics*, **19**, October, 39–41.

Heute U. (1977) Comments on Rabiner L.R. A simplified computational algorithm for implementing FIR digital filters. *IEEE Trans. Acoustics, Speech Signal Processing*, **25**, June, 266–7.

Hillman G.D. (1987) DSP56200: an algorithm-specific digital signal processor peripheral. *Proc. IEEE*, **75**, September, 1185–91.

Knowles J.B. and Olcayto E.M. (1968) Coefficient accuracy and digital filter response. *IEEE Trans. Circuit Theory*, **15**, 31–41.

Lin K., Frantz G.A. and Simar R. (1987) The TMS320 family of digital signal processors. *Proc. IEEE*, **75**, 1143–59.

Lynn P.A. (1970) Economic linear-phase recursive digital filters. *Electronics Lett.*, **6**, 143–5.

Lynn P.A. and Fuerst W. (1989) *Introductory Digital Signal Processing with Computer Applications.* New York: Wiley.

Mintzer F. (1982) On half-band, third-band and *N*th-band FIR filters and their design. *IEEE Trans. Acoustics, Speech Signal Processing*, **30**, 734–8.

Mitra S.K. and Sherwood R.J. (1972) Canonic realizations of digital filters using the continued fraction expansion. *IEEE Trans. Audio Electroacoustics*, **20**, 185–94.

Proakis J.G. and Manolakis D.G. (1992) *Introduction to Digital Signal Processing.* New York: Macmillan.

Rabiner L.R. (1971) Techniques for designing finite-duration impulse response digital filters. *IEEE Trans. Communication Technology*, **19**, 188–95.

Rabiner L.R. (1973) Approximate design relationships for lowpass FIR digital filters. *IEEE Trans. Audio Electroacoustics*, **21**, 456–60.

Rabiner L.R. (1977) A simplified computational algorithm for implementing FIR digital filters. *IEEE Trans. Acoustics, Speech Signal Processing*, **25**, June, 259–61.

Rabiner L.R. and Schafer R.W. (1971) Recursive and nonrecursive realizations of digital filters designed by frequency sampling techniques. *IEEE Trans. Audio Electroacoustics*, **19**, 200–7.

Rabiner L.R. and Schafer R.W. (1972) Correction to 'Recursive and nonrecursive realizations of digital filters designed by frequency sampling techniques'. *IEEE Trans. Audio Electroacoustics (Corresp.)*, **20**, 104–5.

Rabiner L.R., Kaiser J.F. and Schafer R.W. (1974) Some considerations in the design of multiband finite impulse response digital filters. *IEEE Trans. Acoustics, Speech Signal Processing*, **22**(6), 462–72.

Rabiner L.R., McClellan J.H. and Parks T.W. (1975) FIR digital filter design techniques using weighted Chebyshev approximation. *Proc. IEEE*, **63**(4), 595–610.

附录

7A FIR滤波器设计的C语言程序

下面设计FIR滤波器的C语言程序可以在指导手册的CD上找到（参见前言）。

- fresamp.c，使用频率抽样法计算滤波器系数的程序。
- optimal.c，使用最佳方法计算滤波器系数的程序。
- window.c，使用窗口方法计算滤波器系数的程序。
- firfilt.c，对数据进行FIR滤波的程序。
- ncoeff.c，估计最佳低通或者带通滤波器系数个数的程序。

限于篇幅，这里只列出了最后一个程序ncoeff.c的清单（参见程序7A.1）。这个程序是7.6.3节中所给出的方程的直接实现。为了说明这个程序的应用，我们将用它去估计一个带通滤波器的长度，该带通滤波器具有以下规范：

通带	1800 ~ 3300 Hz
阻带	0 ~ 1400, 3700 ~ 5000 Hz
抽样频率	10 kHz
通带波纹	1 dB
阻带衰减	40 dB

程序 7A.1

```
*---------------------------------------------------------------------------*
*                                                                           *
*      program for estimating the number of coefficients of                 *
*      optimal FIR lowpass or bandpass filter                               *
*                                                                           *
*      program name: ncoeff.c
*                                                                           *
*      Manny Ifeachor, 17.10.91                                             *
*                                                                           *
*---------------------------------------------------------------------------*
*/
#include    <stdio.h>
#include    <math.h>
#include    <dos.h>

int       filter__spec();
double    lpfcoeff();
double    bpfcoeff();
float     dp, ds, df;
int       ftype;


main()
{
          double N;
          ftype=filter__spec();                        /* obtain filter specifications */
```

```
        switch(ftype){
              case 1:
                    N=lpfcoeff(); break;
              case 2:
                    N=bpfcoeff(); break;
              default:
                    printf("illegal filter type selected \n");
                    break;
}
        printf("Number of coefficients      \t%f\n",N);
        printf("passband ripple in dB       \t%f\n",dp);
        printf("stopband attenuation in dB  \t%f\n",ds);
        printf("\n");
        printf("press enter to continue \n");
        getch();
        exit(0);
}
/* ------------------------------------------------------------------------------------------------ */
int     filter_spec()
{
        int  itype;
        printf("program to estimate optimal filter length\n");
        printf("\n");
        printf("select filter type\n");
        printf("1    for optimal lowpass filter\n");
        printf("2    for optimal bandpass filter\n");
        scanf ("%d", &itype);
        printf("\n");
        printf("enter passband and stopband deviations in ordinary units\n");
        printf("deviations must be between 0 and 1\n");
        scanf("%f%f",&dp,&ds);
        switch(itype){
              case 1:
                    printf("enter normalized transition width \n");
                    scanf("%f", &df);
                    break;
              case 2:
                    printf("enter normalized transition width - the smaller width\n");
                    scanf("%f", &df);
                    break;
        }
            return(itype);
}
/* ------------------------------------------------------------------------------------------------ */
double  lpfcoeff()
{
        float      ddp, dds, a1, a2, a3, a4, a5, a6, b1, b2;
        double     dinf, ff, t1, t2, t3, t4, Nl;

        /*    constants    */
        a1=0.005309; a2=0.07114; a3=-0.4761; a4=-0.00266;
        a5=-0.5941; a6=-0.4278;
        b1=11.01217; b2=0.5124401;

        ddp=log10(dp);
        dds=log10(ds);
        t1=a1*ddp*ddp;
        t2=a2*ddp;
        t3=a4*ddp*ddp;
```

```
        t4=a5*ddp;
        dinf=((t1+t2+a3)*dds) +(t3+t4+a6);
        ff=b1+b2*(ddp-dds);
        Nl=((dinf/df)-(ff*df)+1);
        dp=20*log10(1+dp); ds=-20*log10(ds);
        return(Nl);
}
/* ------------------------------------------------------------------------------------------------ */
double  bpfcoeff()
{
        float   a1, a2, a3, a4, a5, a6, ddp, dds;
        double  t1, t2, t3, t4, cinf, ginf, Nb;

        a1=0.01201, a2=0.09664, a3=-0.51325; a4=0.00203;
        a5=-0.57054; a6=-0.44314;

        ddp=log10(dp);
        dds=log10(ds);
        t1=a1*ddp*ddp;
        t2=a2*ddp;
        t3=a4*ddp*ddp;
        t4=a5*ddp;
        cinf=dds*(t1+t2+a3)+t3+t4+a6;
        ginf=-14.6*log10(dp/ds)-16.9;
        Nb=(cinf/df) + ginf*df+1;
        dp=20*log10 (1+dp); ds=-20*log10(ds);
        return(Nb);
}
```

根据以上规范，归一化的过渡带宽为0.04 (450/10 000)，由20 log(1+1)得出通带偏差为0.122，由 −20 log (40)得出阻带偏差为0.01。表7A.1中给出了以上例子程序的提示、响应和输出。在这里滤波器系数的个数31仅仅是估计。在大多数的实际应用中，为了满足规范，滤波器长度（也就是滤波器系数个数）要比程序中给出的值高是必要的。在上面的这个例子中，要求的满足规范的实际滤波器长度为35。设计者使用这个程序时应该记住这一点。

表7A.1 ncoeff.c的提示、响应和输出

```
program to estimate optimal filter length

select filter type
1    for optimal lowpass filter
2    for optimal bandpass filter
2

enter passband and stopband deviations in ordinary units deviations must be between 0 and 1
0.122 0.01
enter normalized transition width – the smaller width
0.04
Number of coefficients          31.261084
passband ripple in dB           0.999857
stopband attenuation in dB      40.000000

press enter to continue
```

7B 使用MATLAB进行FIR滤波器设计

MATLAB信号处理工具箱包含了一套优秀的程序和函数，用于设计和分析不同类型的FIR数字滤波器。经由高级命令很容易访问这些程序和命令，使得工具箱成为熟悉FIR的设计和分析而不必陷入复杂的编程的一个很有价值的工具。

在这一节中，我们将举例说明怎样用一些 MATLAB 函数和程序来设计线性相位 FIR 滤波器。特别是，我们将举例说明如何利用窗口方法、最佳方法（Parks-McClellan）和频率抽样方法来计算线性相位滤波器的系数，以及用 MATLAB 来实现上一节讨论的用 C 语言实现的程序。

7B.1　窗口方法

用窗口方法计算标准频率选择、线性相位 FIR 滤波器包含的步骤可以总结如下（详细内容见正文）：

1. 指定期望的频率响应。
2. 选择一个窗函数，并且估计滤波器系数的个数 N。
3. 求理想冲激响应值 $h_D(n)$（截断到 N 个值）。
4. 求窗函数 $w(n)$ 的 N 个系数。
5. 通过应用窗 $h(n) = h_D(n) \times w(n)$，求 FIR 滤波器系数。

对于基于窗的标准频率选择线性相位 FIR 滤波器设计（低通、高通、带通和带阻滤波器），在工具箱中关键的高级命令是 fir1，基本的 fir1 命令的语法是

```
b = fir1(N-1, Fc)
```

这个基本命令是计算并返回具有截止频率 F_c 的 FIR 滤波器的 N 点冲激响应系数。这个命令在 b 向量中返回 N 点系数，b 向量按 z 的负幂的升序排列，

$$b(z) = b(0) + b(1)z^{-1} + b(2)z^{-2} + \dots + b(N-1)z^{-(N-1)}$$

在这个命令中参数 N–1 指定了滤波器的阶数（通常比 FIR 滤波器系数个数小 1）。截止频率 F_c 相对于奈奎斯特频率（即抽样频率的一半）归一化，使其值位于 0 和 1 之间（这里 1 相当于奈奎斯特频率）。

在默认的情况下，基本的 fir1 命令应用哈明窗，并且假定采用的是低通滤波器（如果 F_c 指定的比截止频率还大，则采用带通滤波器）。通过指定滤波器类型和窗函数可以扩展基本命令。在这些情况下，语法为

```
b = fir1(N-1,Fc,'filter-type')
b = fir1(N-1,Fc, window)
b = fir1(N-1,Fc,'filter-type', window)
```

对于一个高通滤波器，单字 “high” 指定滤波器的类型；对于带阻来说，采用单字 “stop”。对于带通滤波器和带阻滤波器，变量 F_c 都是一个定义截止频率的矢量；对于高通和带阻滤波器，滤波器长度必须是奇整数（偶整数不适合于高通和带阻滤波器，因为它们会在前面描述的奈奎斯特频率处产生一个零幅值响应）。

MATLAB 支持众多窗函数的应用，包括哈明窗、汉宁窗、方脉冲窗（矩形窗）、凯塞窗和切比雪夫窗。生成窗系数的语法是

```
w = boxcar(N)
w = blackman(N)
w = hamming(N)
w = hanning(N)
w = kaiser(N, beta)
```

实际应用中，窗命令常常嵌入到 fir1 命令中（参见后面的例子）。

应该指出的是，由于实现上的不同，使用 MATLAB 设计基于窗的 FIR 滤波器时，与其他程序相比可能会得到不同的结果。例如，在 MATAB 中，加窗后冲激响应系数可能会伸缩，使得在通带的中间给出单位幅度-频率响应。可以加上 “noscale” 来忽略这一点，即 b = fir1(N–1, Fc, ‘noscale’)。

而且，大部分窗函数的MATLAB实现和先前的实现可能有点不同，这也导致结果上的微小差异。设计者应该意识到这些差别，如果有必要，可以对这些细微差别做适当的修正。

例7B.1 求线性相位FIR低通滤波器的系数，它的通带带沿频率和阻带带沿频率分别是1 kHz和4.3 kHz。采用哈明窗，并假设抽样频率为10 kHz。

如表7.3所示，对于基于哈明窗的滤波器，过渡带宽Δf与滤波器长度N有如下近似关系：

$$N \approx \frac{3.3}{\Delta f}$$

现在，Δf为0.33（根据（4.3−1）/10），滤波器长度$N = 10$。根据正文中的方法，实际截止频率（允许有拖尾效应）取通带带沿频率和阻带带沿频率的中间值，即2.65 kHz。在MATLAB中，截止频率用抽样频率的一半做归一化，即f_c(归一化) = 2.65/5 = 0.53。

在程序7B.1中给出了MATLAB命令。截断的理想的冲激响应值、窗和加窗的滤波器的系数在表7B.1给出。

程序7B.1 计算例7B.1的FIR滤波器系数的MATLAB m文件

```
fc=0.53;                       % Cutoff frequency (normalized to Fs/2)
N=10;                          % Filter length (number of taps)
hd=fir1(N-1,fc,boxcar(N));     % Truncated ideal impulse response
wn=hamming(N);                 % Calculate Hamming window coefficients
hn=fir1(N-1,fc,wn);            % Obtain windowed coefficients
```

表7B.1 例7B.1的滤波器参数

n	截断的理想冲激响应 $hD(n)$	窗系数 $w(n)$	加窗的滤波器系数 $h(n)$
0	0.0641	0.0800	0.0053
1	−0.0388	0.1876	−0.0075
2	−0.1052	0.4601	−0.0498
3	0.1235	0.7700	0.0974
4	0.4564	0.9723	0.4544
5	0.4564	0.9723	0.4544
6	0.1235	0.7700	0.0974
7	−0.1052	0.4601	−0.0496
8	−0.0388	0.1876	−0.0075
9	0.0641	0.0800	0.0053

例7B.2 举例说明使用凯塞窗计算FIR滤波器系数。利用凯塞窗和MATLAB求满足下列规范的带通FIR滤波器系数并画出幅度－频率响应：

通带	150～250 Hz
过渡带宽	50 Hz
阻带波纹	0.1 dB
阻带衰减	60 dB
抽样频率	1 kHz

解：

这个问题和正文中的例7.4相同，我们将用MATLAB来求解。

例7.4给出的滤波器的长度$N = 73$，波纹系数$\beta = 5.65$。程序7B.2为MATLAB程序。滤波器系数和幅度谱如表7B.2和图7B.1所示。

在这个例子中我们应该注意到，在fir1命令中已经包括了窗的类型来取代单独计算窗口系数。

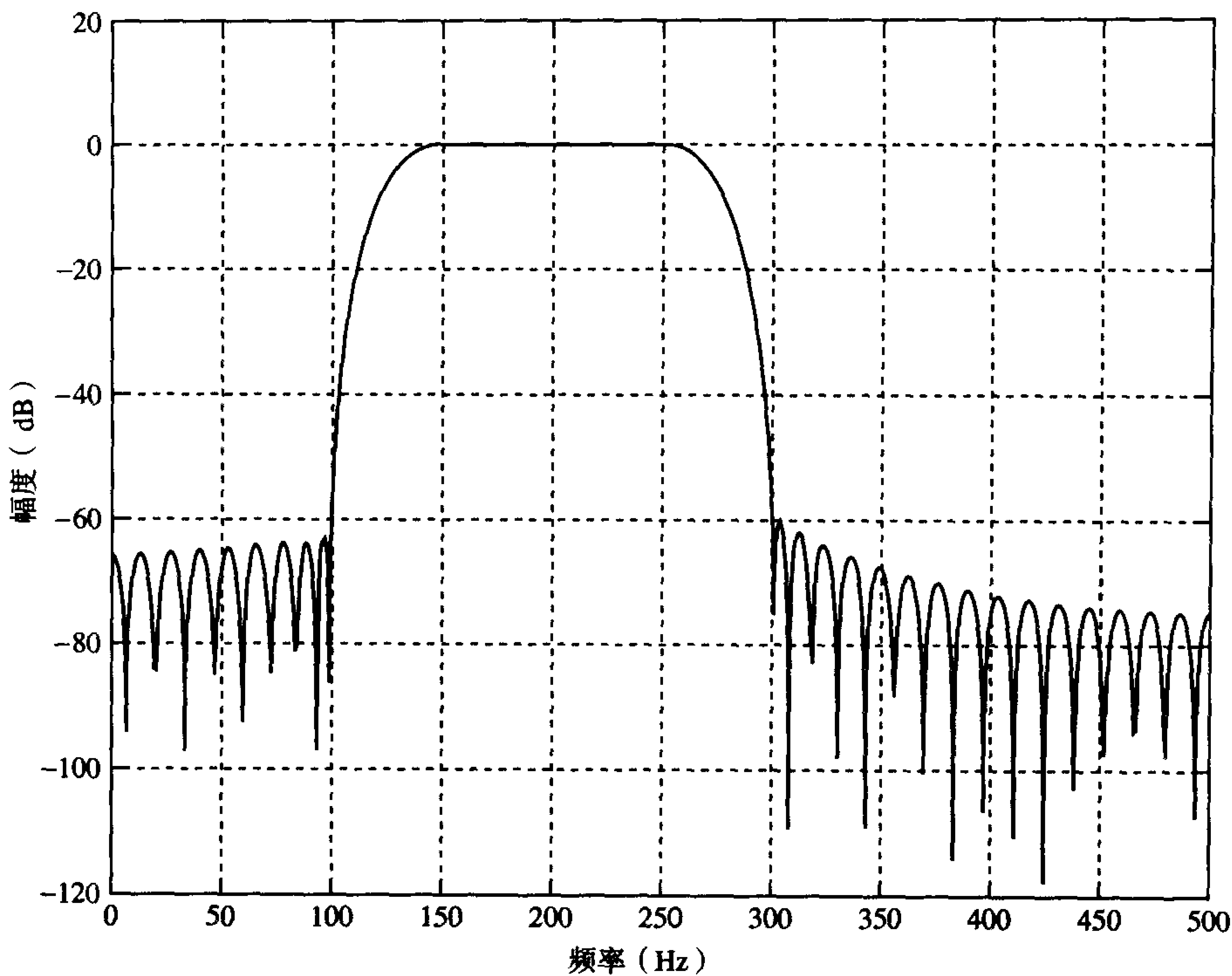


图 7B.1　例 7B.2 的幅度谱

程序 7B.2　计算例 7B.2 的 FIR 滤波器系数的 MATLAB m 文件

```
FS=1000;                                  % Sampling frequency
FN=FS/2;                                  % Nyquist frequency
N=73;                                     % Filter length
beta=5.65;                                % Kaiser window Ripple parameter
fc1=125/FN;                               % Normalized cut off frequencies
fc2=275/FN;
FC=[fc1 fc2];                             % Band edge frequency vector
hn=fir1(N–1, FC, kaiser(N, beta));        % Obtain windowed filter coeffs
[H, f]=freqz(hn, 1, 512, FS);             % Compute frequency response
mag=20*log10(abs(H));
plot (f, mag), grid on
xlabel ('Frequency (Hz)')
ylabel ('Magnitude Response (dB)')
```

表 7B.2　例 7B.2 的滤波器系数

n	$h(n)$
0	−0.0001
1	−0.0004
2	−0.0001
3	−0.0001
4	−0.0007
5	0.0005
6	0.0023
7	0.0008
8	−0.0017
9	−0.0005
10	−0.0005
11	−0.0044
12	−0.0022
13	0.0069
14	0.0066
15	−0.0016

（续表）

n	h(n)
16	0.0000
17	0.0022
18	−0.0117
19	−0.0164
20	0.0069
21	0.0189
22	0.0029
23	0.0044
24	0.0188
25	−0.0125
26	−0.0520
27	−0.0165
28	0.0333
29	0.0104
30	0.0094
31	0.0856
32	0.0453
33	−0.1665
34	−0.2066
35	0.0891
36	0.2998

7B.2 最佳方法

在MATLAB的信号处理工具箱中，包含了许多设计程序和基于Park-McClellan和Remez算法的设计FIR滤波器的函数。remez命令是通过最佳方法计算FIR系数的关键命令。这个命令可以用来设计多频带线性相位FIR滤波器，在它的基本形式中，命令具有下面的语法：

```
b = remez(N−1, F, M)
```

式中N是滤波器长度，F是归一化的带沿频率的矢量，M是在指定的带沿频率处滤波器期望的幅度响应的矢量。带沿频率用半抽样频率归一化，并位于0和1之间（奈奎斯特频率对应于1）。

这个基本命令通过特殊指定可以将其扩展。例如，指定通带、阻带内的相对权值和滤波器类型。在这种相对权值是指定的和要求通带和阻带的偏差的情况下，命令的语法是

```
b = remez(N−1, F, M, WT)
```

式中WT是频带内波纹之间的相对权值矢量。

也可以加上标记‘ftype’来指定要求的滤波器类型。滤波器有四种可能的类型，取决于N是奇数还是偶数以及滤波器系数的对称类型。当滤波器长度N为奇数时（就是当N−1是偶数时），滤波器为类型1；当滤波器长度为偶数时，滤波器为类型2。标准频率选择滤波器对滤波器类型1的应用没有限制。类型2滤波器在奈奎斯特频率上有一个零点，因此它不能用来设计高通和阻带滤波器。类型3（N是奇数）滤波器导致希尔波特变换，而类型4（N是偶数）为差分器。对于类型1和类型2滤波器，指定长度就足以表明选择，但是对类型3和类型4，必须包含标记‘hilbert’或者‘differentiator’来表示滤波器的类型。

例7B.3 用最佳算法计算滤波器系数，并画出满足下面特性的带通线性相位FIR滤波器的频率响应：

通带	1000～1500 Hz
过渡带宽	500 Hz
滤波器长度	41
抽样频率	10 000 Hz

解：

滤波器的频带为 0 ~ 500 Hz（低阻带），1000 ~ 1500 Hz（通带），2000 ~ 5000 Hz（高阻带）。带沿频率必须用半抽样频率归一化：

$$500/5000 = 0.1$$
$$1000/5000 = 0.2$$
$$1500/5000 = 0.3$$
$$2000/5000 = 0.4$$
$$5000/5000 = 1$$

因此规一化的带沿频率矢量 F 变为

$$F \quad [0, 0.1, 0.2, 0.3, 0.4\ 1]$$

期望的幅度响应在通带内是 1，而在阻带内是 0，给定期望的幅度响应矢量：

$$M = [0\ 0\ 1\ 1\ 0\ 0]$$

程序 7B.3 给出了用于计算滤波器系数和绘制滤波器幅度响应的 MATLAB 命令。

滤波器系数和滤波器的幅度响应如表 7B.3 和图 7B.2 所示。

程序 7B.3　计算最佳 FIR 滤波器系数及绘制频率响应的 MATLAB m 文件

```
%
%
Fs=10000;                      % Sampling frequency
N=41;                          % Filter length
M=[0 0 1 1 0 0];               % Desired magnitude response
F=[0, 0.1, 0.2, 0.3, 0.4 1];   % Band edge frequencies
b = remez(N–1, F, M);          % Compute the filter coefficients
[H, f] = freqz(b, 1, 512, Fs); % Compute the frequency response
mag = 20*log10(abs(H));        % of filter and plot it
plot(f, mag)
xlabel('Frequency (Hz)')
ylabel('Magnitude (dB)')
```

表 7B.3　例 7B.2 的滤波器系数

n	$h(n)$
0	−0.0001
1	−0.0004
2	−0.0001
3	−0.0001
4	−0.0007
5	0.0005
6	0.0023
7	0.0008
8	−0.0017
9	−0.0005
10	−0.0005
11	−0.0044
12	−0.0022
13	0.0069
14	0.0066
15	−0.0016
16	0.0000
17	0.0022
18	−0.0117
19	−0.0164
20	0.0069
21	0.0189

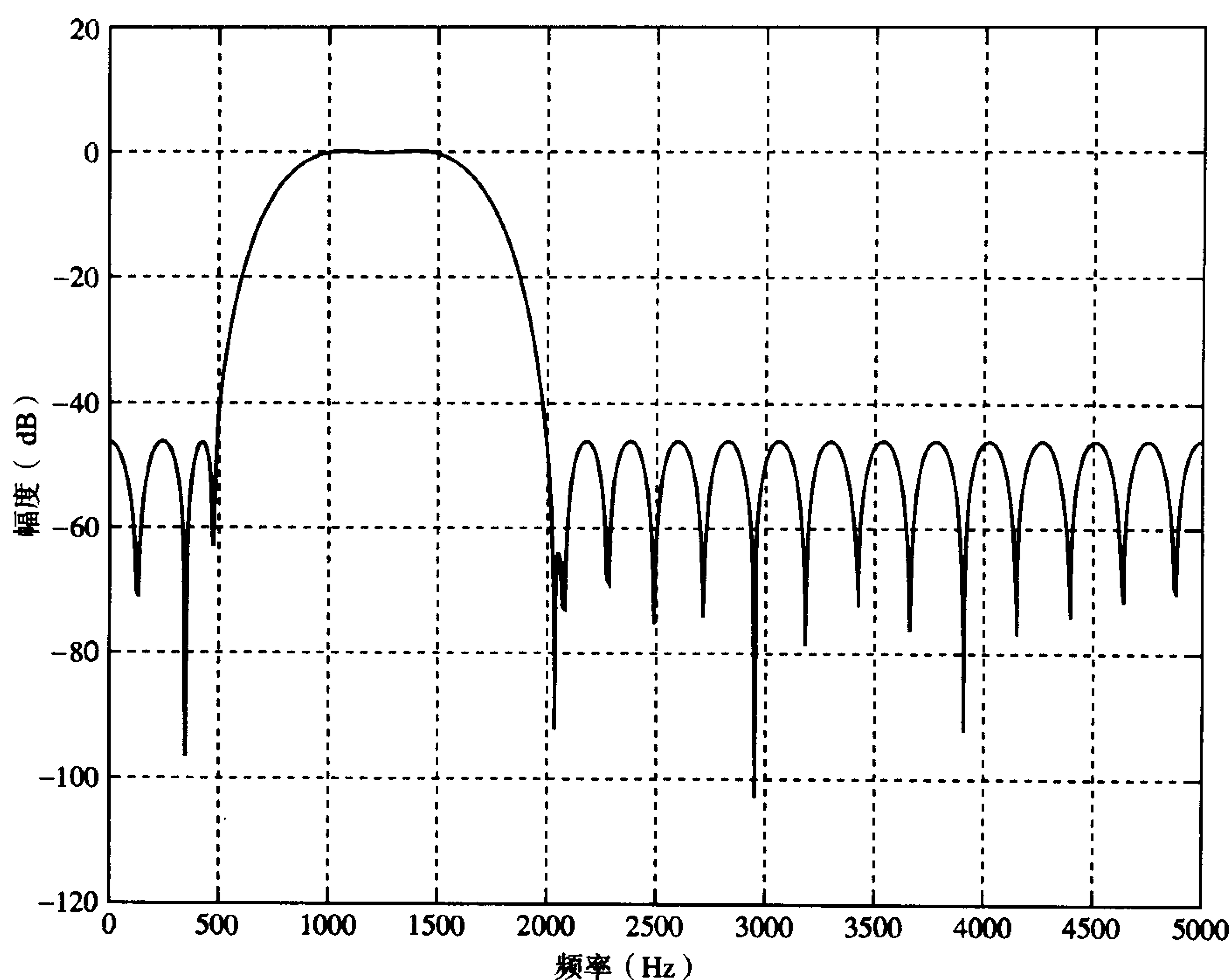


图 7B.2　例 7B.3 的幅度响应

例 7B.4　设计一个线性相位带通滤波器，要求满足下面这解参数：

通带	12 ~ 16 kHz
过渡带宽	2 kHz
通带波纹	1 dB
阻带衰减	45 dB
抽样频率	50 kHz

估算滤波器长度N，用最佳方法确定滤波器系数，然后绘出幅度－频率响应图。将滤波器的通带和阻带波纹与指定的值进行比较。

解：

正如前面所述，带沿频率需要相对于奈奎斯特频率做归一化：

$$10/25 = 0.4$$
$$12/25 = 0.48$$
$$16/25 = 0.64$$
$$18/25 = 0.72$$

使用这些值求得带沿频率矢量是

```
F = [0 0.4 0.48 0.64 0.72 1]
M = [0 0 1 1 0 0]
```

滤波器长度可以用 remezord 命令来计算。这要求通带和阻带波纹用标准的线性单位表示。因此，我们从分贝换算这些值：

$$\delta_p = \frac{10^{\frac{A_p}{20}} - 1}{10^{\frac{A_p}{20}} + 1}, \quad \delta_s = 10^{\frac{-A_s}{20}}$$

式中 A_p 和 A_s 分别表示通带和阻带的波纹，单位是 dB。

带沿频率、期望的幅度响应和波纹值以及抽样频率都用来求滤波器阶数(N−1)的估计值以及滤波器长度（参见程序 7B.4a、图 7B.3 和表 7B.4）。

滤波器参数估值：$N=40$，权值 10.22 : 1 : 10.22，最大偏差 0.0774。可以增大 N 值，以达到更大的阻带衰减或者更低的通带波纹（参见程序 7B.4b）。

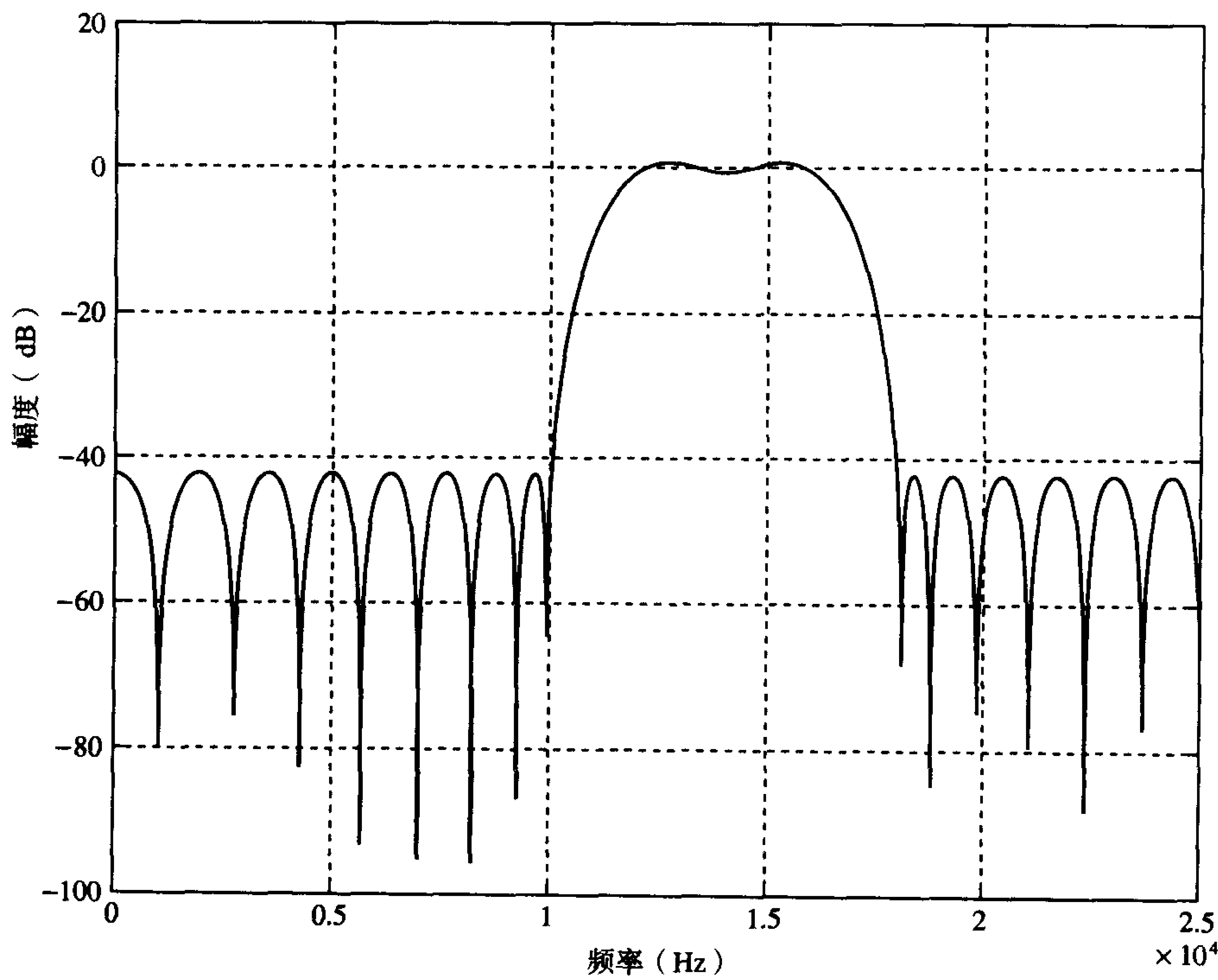


图 7B.3　例 7B.4 的幅度响应

表 7B.4　例 7B.4 中最佳 FIR 滤波器的系数

n	h(n)
0	0.0005
1	−0.0017
2	−0.0088
3	0.0139
4	0.0136
5	−0.0273
6	−0.0060
7	0.0363
8	−0.0059
9	−0.0225
10	0.0054
11	−0.0080
12	0.0305
13	0.0293
14	−0.0988
15	−0.0085

（续表）

n	h(n)
16	0.1654
17	−0.0595
18	−0.1854
19	0.1411

程序 7B.4a 例 7B.4 中用于计算最佳 FIR 滤波器系数和绘制频率响应的 MATLAB m 文件

```
%
%
Fs=50000;                                    % Sampling frequency
Ap=1;                                        % Pass band ripple in dB
As=45;                                       % Stop band attenuation in dB
M=[0 1 0];                                   % Desired magnitude response
F=[10000, 12000, 16000, 18000];              % Band edge frequencies
dp=(10^(Ap/20)-1)/(10^(Ap/20)+1);            % Pass and stop band ripples
ds=10^(-As/20);
dev=[ds dp ds];
[N1, F0, M0, W] = remezord(F, M, dev, Fs)    % Determine filter order
[b delta] = remez(N1, F0, M0, W);            % Compute the filter coefficients
[H, f] = freqz(b, 1, 1024, Fs);              % Compute the frequency response
mag = 20*log10(abs(H));                      % of filter and plot it
plot(f, mag), grid on
xlabel('Frequency (Hz)')
ylabel('Magnitude (dB)')
```

程序 7B.4b 例 7B.4 中一个计算最佳 FIR 滤波器系数并绘制频率响应的可选的 MATLAB m 文件

```
%
%
N=44
Fs=50000;                               % Sampling frequency
Ap=1;                                   % Pass band ripple in dB
As=45;                                  % Stop band attenuation in dB
M=[0 0 1 1 0 0];                        % Desired magnitude response
F=[0, 0.4, 0.48, 0.64, 0.72 1] ;        % Band edge frequencies
dp=(10^(Ap/20)-1)/(10^(Ap/20)+1);
ds=10^(-As/20);
W=[dp/ds, 1, dp/ds];
dev=[ds ds dp dp ds ds];
[b delta] = remez(N-1, F, M, W);        % Compute the filter coefficients
[H, f] = freqz(b, 1, 1024, Fs);         % Compute the frequency response
mag = 20*log10(abs(H));                 % of filter and plot it
plot(f, mag), grid on
xlabel('Frequency (Hz)')
ylabel('Magnitude (dB)')
```

7B.3 频率抽样方法

fir2 命令用来设计具有任意频率响应特性（比如在频率抽样方法中遇到的）的 FIR 滤波器。这种基本命令的语法是

```
b = fir2(N-1, F, H)
```

fir2 命令计算长度为 N 的 FIR 滤波器的系数。矢量 F 指定范围在 0 和 1 之间的归一化频率点（其中频率点如以前那样用半抽样频率做归一化）。矢量 H 具体指定了由 F 指定的频率点的期望的幅度响应。这两个矢量具有相同的长度。

我们将给出一些例子来说明使用 fir2 命令的 FIR 滤波器设计。

例7B.5　频率抽样的滤波器设计举例　一个线性相位频率抽样FIR滤波器有两个过渡频率抽样值，假设滤波器有 15 个抽头，滤波器由下面的频率抽样值所描述：

$$
\begin{aligned}
|H(k)| &= 1 & k &= 0, 1, 2, 3 \\
&0.5571 & k &= 4 \\
&0.0841 & k &= 5 \\
&0 & k &= 6, 7
\end{aligned}
$$

如果抽样频率为 2 kHz，确定滤波器系数值。

解：

以上的频率抽样值已经指定了频率范围在0到半抽样频率之间。因此，用半抽样频率归一化后的频率点为：0, 1/7, 2/7, 3/7, 4/7, 5/7, 6/7, 1。

程序 7B.5 给出了用频率抽样值确定 FIR 滤波器系数的 MATLAB 程序，图 7B.4 描述了滤波器的幅度－频率响应，表 7B.5 列出了滤波器系数。

程序 7B.5　用于计算 FIR 频率抽样滤波器系数的 MATLAB m 文件

```
N=15;
fd=[0 1/7 2/7 3/7 4/7 5/7 8/7 1];
Hd=[1 1 1 1 0.5571 0.0841 0 0];
hn=fir2(N-1, fd, Hd);
[H, f] = freqz(hn, 1, 512, Fs);
plot(f, abs(H)), grid on
xlabel('Frequency (Hz)')
ylabel('Magnitude')
```

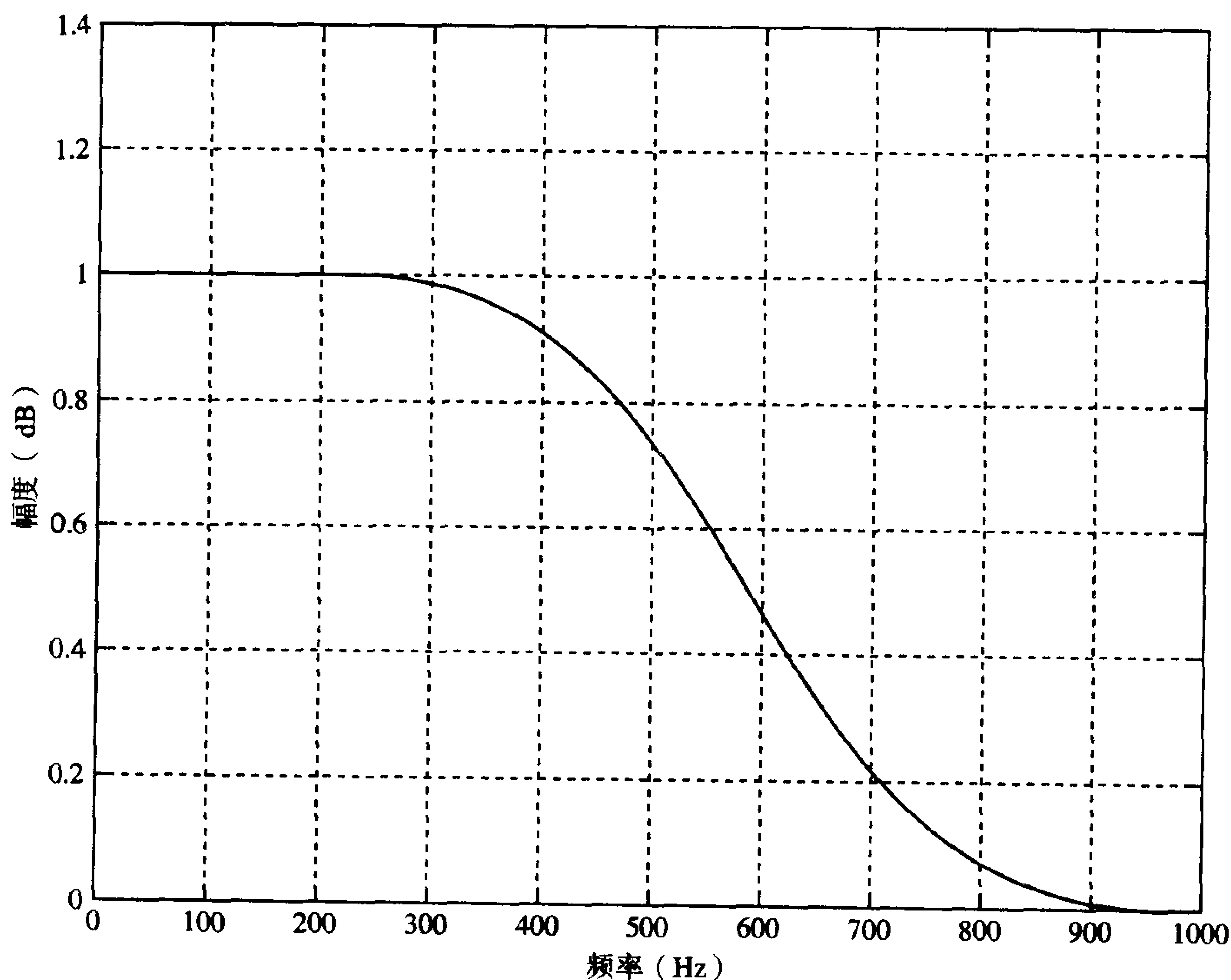


图 7B.4　例 7B.5 的幅度－频率响应

表 7B.5 例 7B.5 中的滤波器系数

n	$h(n)$
0	−0.0001
1	−0.0006
2	0.0017
3	0.0128
4	−0.0299
5	−0.0571
6	0.2777
7	0.5910
8	0.2777
9	−0.0571
10	−0.0299
11	0.0128
12	0.0017
13	−0.0006
14	−0.0001

例 7B.6 具有任意幅度响应滤波器的设计 设计一个 FIR 滤波器，它近似于图 7B.5 描述的幅度－频率响应特性。

确定合适的 FIR 滤波器的系数值，绘出该滤波器的幅度－频率响应。假设抽样频率为 2 kHz，滤波器长度为 110。

解：

所期望的幅度响应值在归一化频率 0 ~ 0.15 之间为 1，在 0.25 ~ 0.45 之间为 0.2，在 0.5 ~ 0.75 之间为 0.1，在 0.85 和 1 之间为 0。在 MATLAB 程序中必须指定这些值和归一化频率。

程序 7B.6 列出了具有指定的频率抽样的 MATLAB 程序。使用 fir2 命令计算滤波器系数，使用 freqz 命令绘出滤波器的幅度响应。因篇幅有限，这里就不列出滤波器系数了。图 7B.6 描述的是 FIR 滤波器的幅度－频率响应。

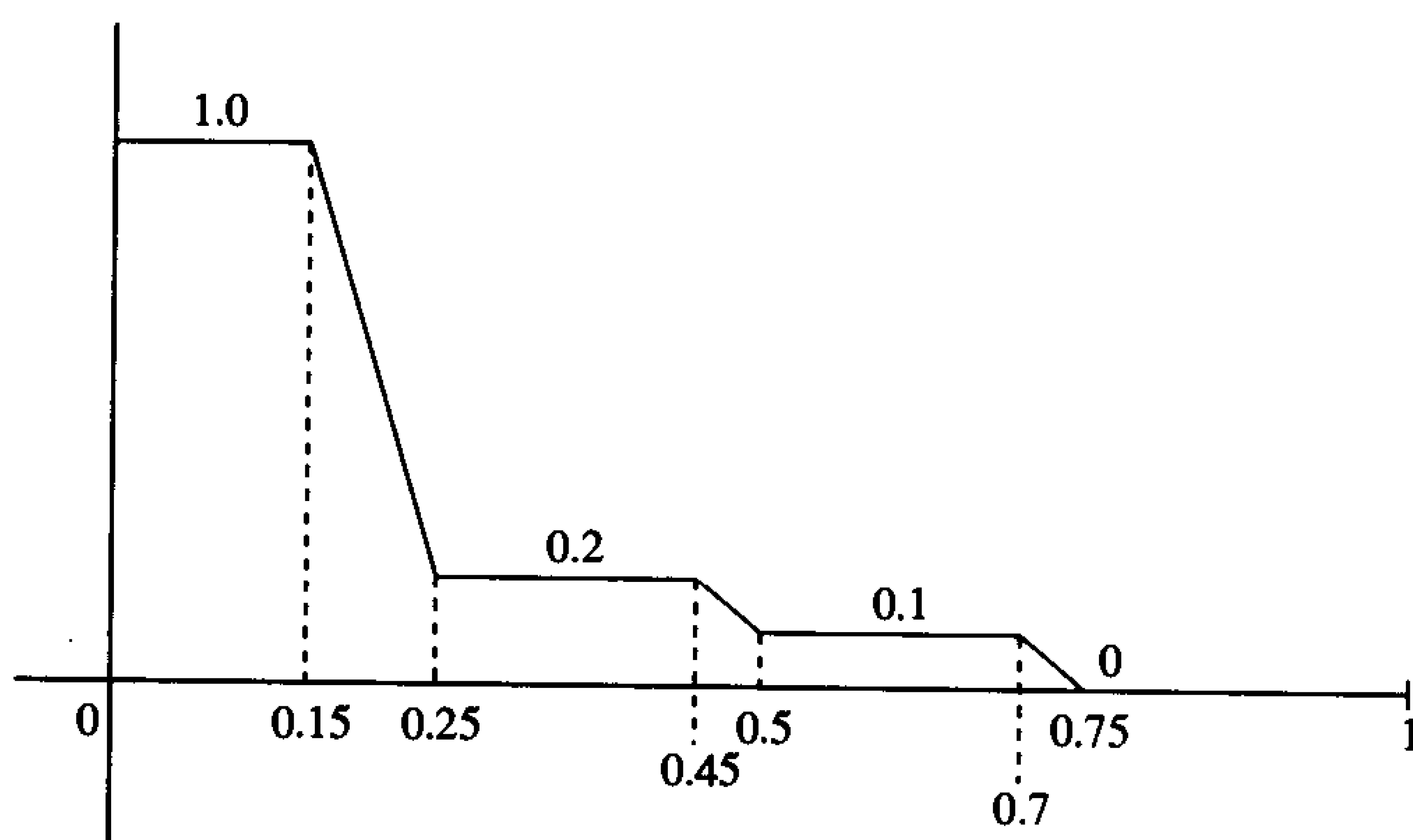


图 7B.5 例 7B.6 的幅度－频率响应特性

程序 7B.6 计算一个具有任意幅度响应的 FIR 滤波器的系数的 MATLAB m 文件

```
Fs=2000;                                  % Sampling frequency
N=110;                                    % Filter length
fd=[0 0.15 0.25 0.45 0.5 0.75 0.85 1];    % Frequency sampling points
Hd=[1 1 0.3 0.3 0.1 0.1 0 0];             % Frequency samples
hn=fir2(N-1, fd, Hd);                     % Compute the impulse response
[H, f] = freqz(hn, 1, 512, Fs);
plot(f, abs(H)), grid on
xlabel('Frequency (Hz)')
ylabel('Magnitude')
```

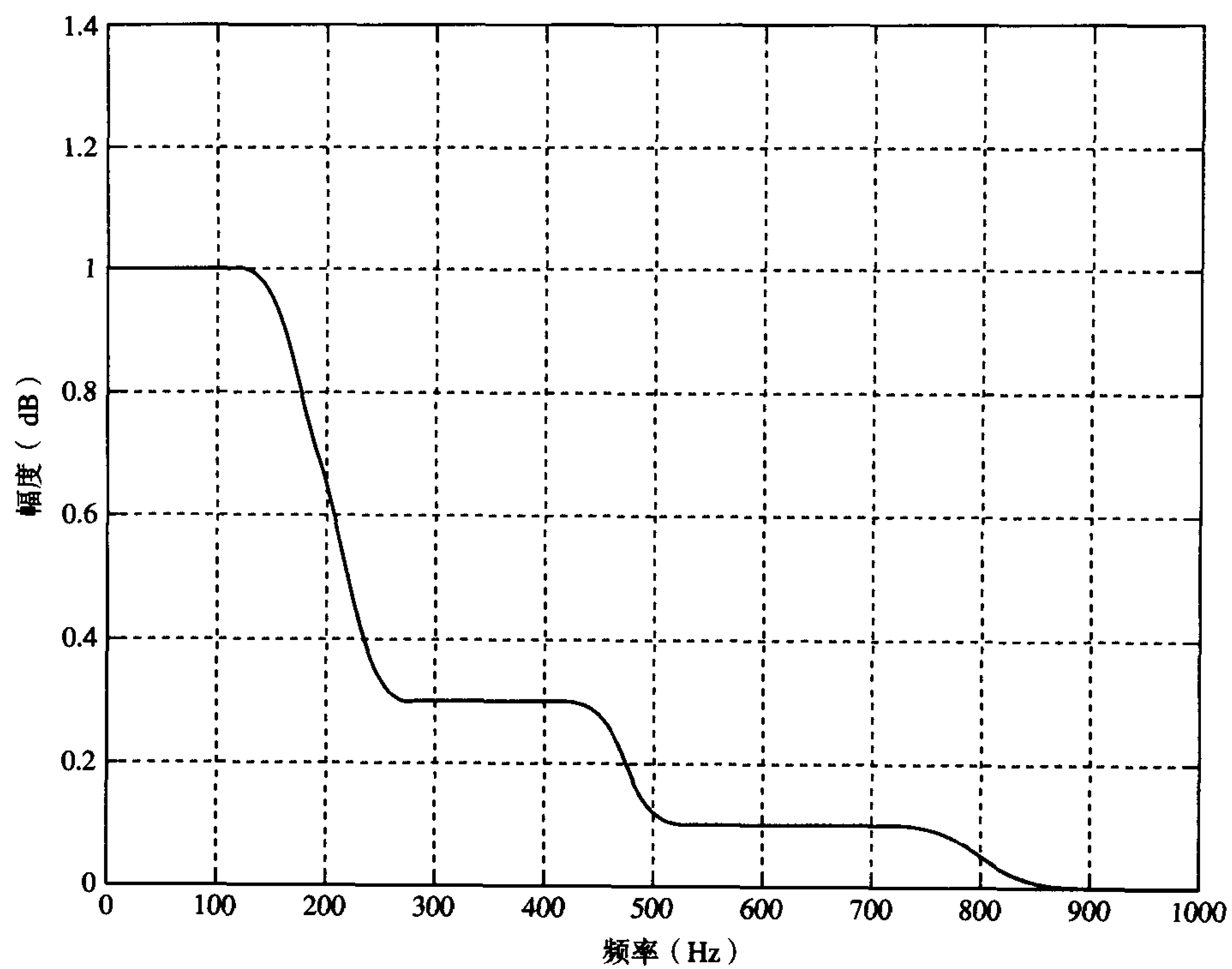


图 7B.6　例 7B.6 设计的 FIR 滤波器的幅度－频率响应

第8章　无限冲激响应（IIR）数字滤波器的设计

本章介绍了无限冲激响应（IIR）数字滤波器的实用设计方法，包括通用的方法，该方法允许将模拟滤波器转化成等价的数字滤波器。本章还描述了一个简单却通用的从技术规范到实现的逐步指南来设计IIR滤波器。我们给出了几个已处理过的例子来说明数字IIR滤波器设计的各个方面，包括有限精度算法对滤波器性能和实时实现的影响分析。

我们提供了许多MATLAB和C语言程序来使读者能够计算滤波器系数且进行有限字长分析。滤波器设计的通用结构的描述、IIR和FIR的比较以及数字和模拟滤波器的比较，请读者参看第6章。在本章中我们将集中讲述IIR滤波器的设计和应用。

8.1　引言：IIR滤波器基本特征概要

可实现的IIR数字滤波器是通过下面的递归方程来刻画的：

$$y(n)=\sum_{k=0}^{\infty}h(k)x(n-k)=\sum_{k=0}^{N}b_kx(n-k)-\sum_{k=1}^{M}a_ky(n-k) \tag{8.1}$$

其中$h(k)$是滤波器的冲激响应，理论上它的持续时间是无限的。b_k和a_k是滤波器的系数，$x(n)$和$y(n)$是滤波器的输入和输出。IIR滤波器的传递函数为

$$H(z)=\frac{b_0+b_1z^{-1}+\ldots+b_Nz^{-N}}{1+a_1z^{-1}+\ldots+a_Mz^{-M}}=\frac{\sum_{k=0}^{N}b_kz^{-k}}{1+\sum_{k=1}^{M}a_kz^{-k}} \tag{8.2}$$

IIR滤波器设计过程的一个重要部分是对系数a_k和b_k求出适当的值，使得滤波器的某些特性（如频率响应等）可以表现为期望的方式。8.1式和8.2式是IIR滤波器的特征方程。

注意，在8.1式中，当前的输出抽样值$y(n)$是过去输出$y(n-k)$以及当前和过去输入的抽样值$x(n-k)$的函数，也就是IIR滤波器是某种类型的反馈系统。IIR滤波器的优势源于反馈系统提供的灵活性。例如，对于同样的技术规范，IIR滤波器要求的系数通常比FIR滤波器少，这也就是当重点要求锐截止和高吞吐率时为什么要使用IIR滤波器的原因。这样的代价是，IIR滤波器可能变得不稳定，或者在设计时没充分考虑细节，那么它的性能会显著降低。

8.2式的IIR滤波器的传递函数$H(z)$可以因式分解为

$$H(z)=\frac{K(z-z_1)(z-z_2)\ldots(z-z_N)}{(z-p_1)(z-p_2)\ldots(z-p_M)} \tag{8.3}$$

其中$z_1, z_2, \ldots$是$H(z)$的零点，也就是那些使$H(z)$变成零的z值，$p_1, p_2, \ldots$是$H(z)$的极点，也就是使$H(z)$变成无穷大的z值。

传递函数的极点和零点的图称为极零图，它提供了一个在复z平面表示和分析滤波器的非常有用的方法；细节请参见第3章。为了使滤波器稳定，它的所有极点必须位于单位圆内（或者与零点恰好在单位圆上重合），对零点的位置则没有限制。

8.2　数字 IIR 滤波器的设计步骤

IIR 滤波器的设计可以很方便地分解为如下五个主要步骤：

(1) 滤波器的性能规范，在这个阶段设计者给出滤波器的函数（例如低通）及所期望的性能。
(2) 近似或系数计算，这时我们从许多方法中选择一种来计算传递函数 $H(z)$ 中的系数 a_k 和 b_k 的值，使步骤 1 中给出的规范能够满足。
(3) 实现结构，这一步就是简单地将传递函数转化成一个合适的滤波器结构。IIR 滤波器的典型结构是二阶和/或一阶滤波器单元的并联和串联。
(4) 误差分析，误差是源于滤波器系数的表达以及由算术运算采用有限位数而产生的。
(5) 实现，这里包括硬件构建和软件代码的编写、执行实际的滤波运算。

这些步骤在图 8.1 中进行了概括。正如图中所指出的那样，五个步骤不是独立的，它们并不总是按照给定的顺序执行。实际上，常常是将第二、第三和第五步组合起来。然而，这里讨论的方法可以确保设计成功。为了得到一个高效的滤波器，在每一步内或者这些步骤之间重复多次可能是很有必要的。

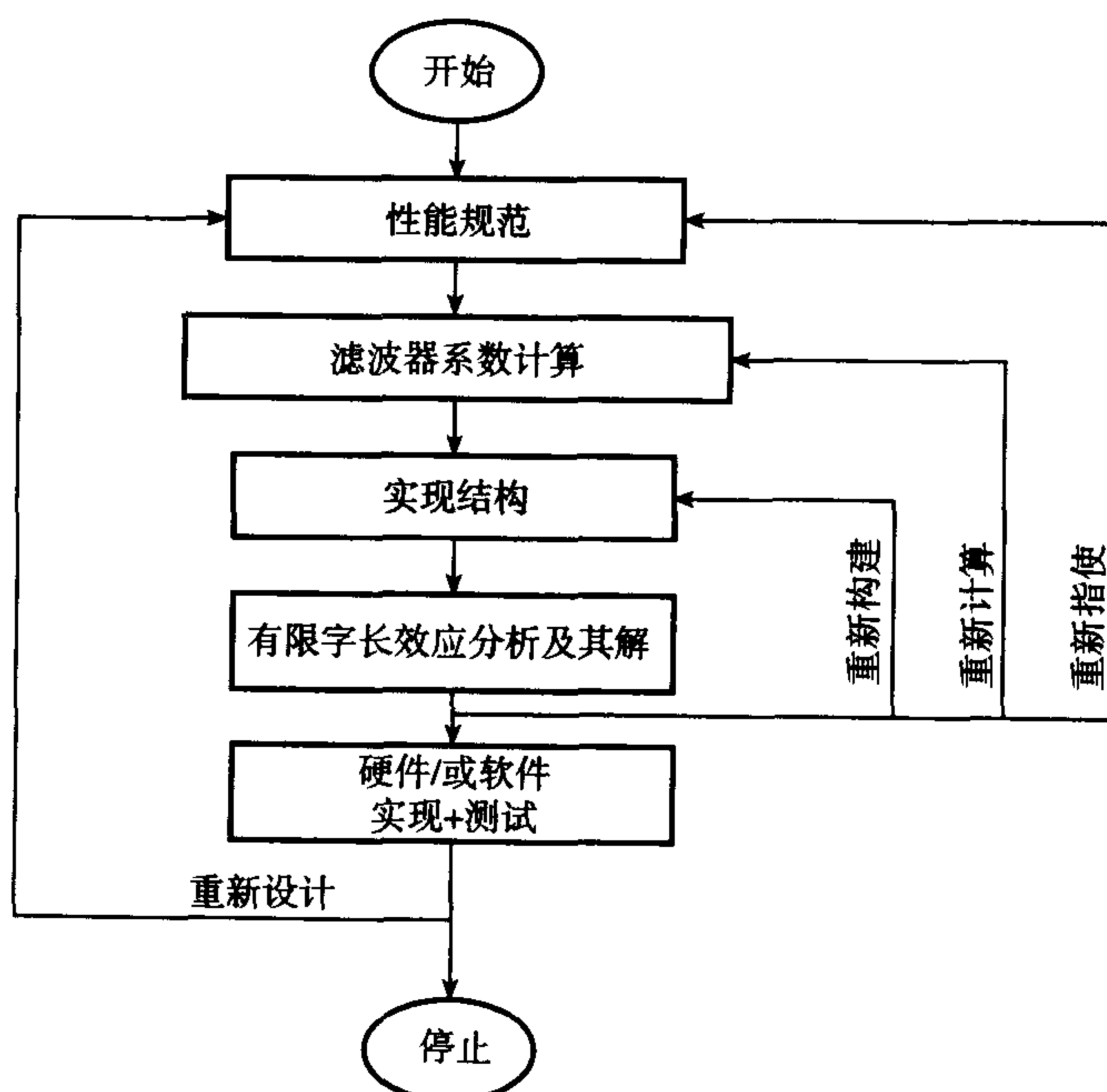


图 8.1　数字滤波器设计步骤概要

8.3　性能规范

如大多数其他工程问题一样，数字 IIR 滤波器从一份清晰的性能要求规范开始。这些要求包括 (i)信号特性（信号源及接收器的类型，I/O 接口，数据率和字长，感兴趣的频率）；(ii)滤波器的频率响应特性（期望的幅度和相位响应及它们的容差，运算速度）；(iii)实现方法（例如，在计算机中的高级语言程序或者基于 DSP 处理器的系统，信号处理器的选择，滤波模式的选择——实时或批处理）；(iv)其他的设计约束（例如成本、通过滤波器允许的信号衰减）。一般来说，以上的大部分要求是与应用有关的。设计者可能没有足够的信息在一开始就完全确定滤波器的规范，但是应该给出尽可能多的滤波器要求来简化设计过程。

对于频率选择性滤波器，例如低通和带通滤波器，频率响应的规定通常是用容差图的形式给出。图8.2对带通IIR滤波器画出了一个这样的图。水平阴影部分指示了容差限制。下面的参数通常用来刻画频率响应。

ε^2	通带波纹参数
δ_p	通带偏差
δ_s	阻带偏差
f_{p1} 和 f_{p2}	通带边沿频率
f_{s1} 和 f_{s2}	阻带边沿频率

某些时候通带的边沿频率是用归一化的形式给出的，它是相对于抽样频率的分式（f/F_s）。但是，我们应该用标准的频率单位赫兹或千赫来定义它，特别是对于不是很有经验的设计者，这样就不会产生混淆。通带或阻带的偏移可以用普通的数字或分贝表示，通带波纹用分贝表示为

$$A_p = 10\log_{10}(1+\varepsilon^2) = -20\log_{10}(1-\delta_p) \tag{8.4a}$$

阻带衰减用分贝表示为

$$A_s = -20\log_{10}(\delta_s) \tag{8.4b}$$

如第6章讨论过的那样，以及从图8.2中可以很明显看出，IIR滤波器的通带波纹是通带间最大值和最小值的偏差值。对于FIR滤波器，通带波纹是通带间理想响应和最大值（或最小值）的偏差值。因此，对于IIR，当我们说起通带时，我们指的是峰峰通带波纹。

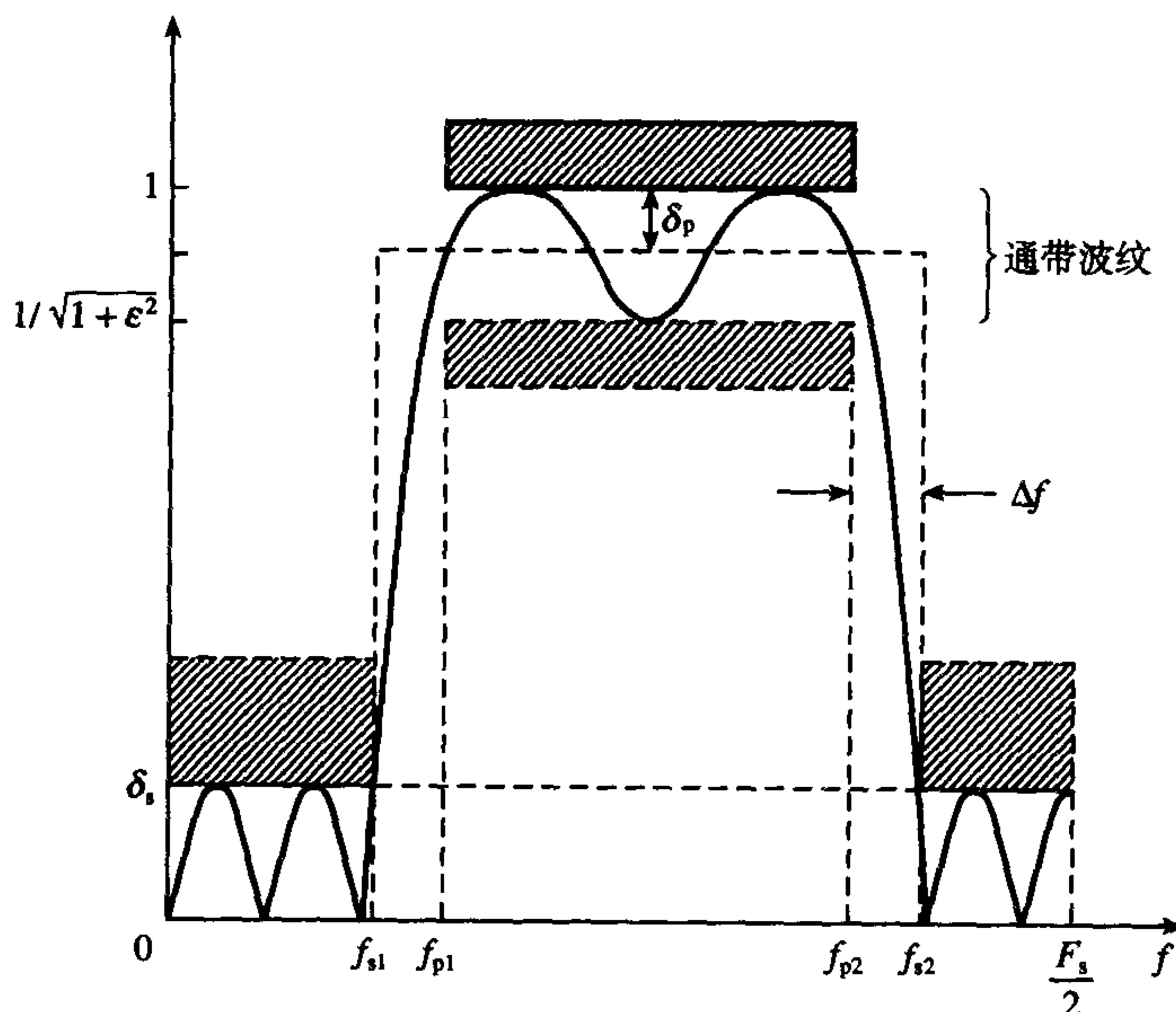


图8.2 IIR带通滤波器的容差图

8.4 IIR滤波器的系数计算方法

本阶段的任务是在众多的近似方法中选择一个，并且用它来计算8.2式中的系数 a_k 和 b_k，使得第一个设计阶段给出的频率响应的规范能够满足。

求IIR滤波器系数的一个简单方法是在z平面适当地放置极点和零点，使得到的滤波器具有期望的频率响应。这个方法称之为极－零点放置方法，这一方法仅对非常简单的滤波器才有用，例如陷波滤波，它的滤波器参数（例如通带波纹）不需要很精确地指定。一个更有效的方法是首先设计满足期望规定的模拟滤波器，然后把它转化成一个等价的数字滤波器。大部分的IIR数字滤波器都是用这种方法设计的。这种方法的合理性在于在文献中我们已经有了大量的可利用的模拟滤波器的信息。三种最基本的把模拟滤波器转化成等价数字滤波器的方法是冲激不变法、匹配z变换法和双线性z变换法。

在下一节将涵盖如下几个计算 IIR 滤波器系数的方法：

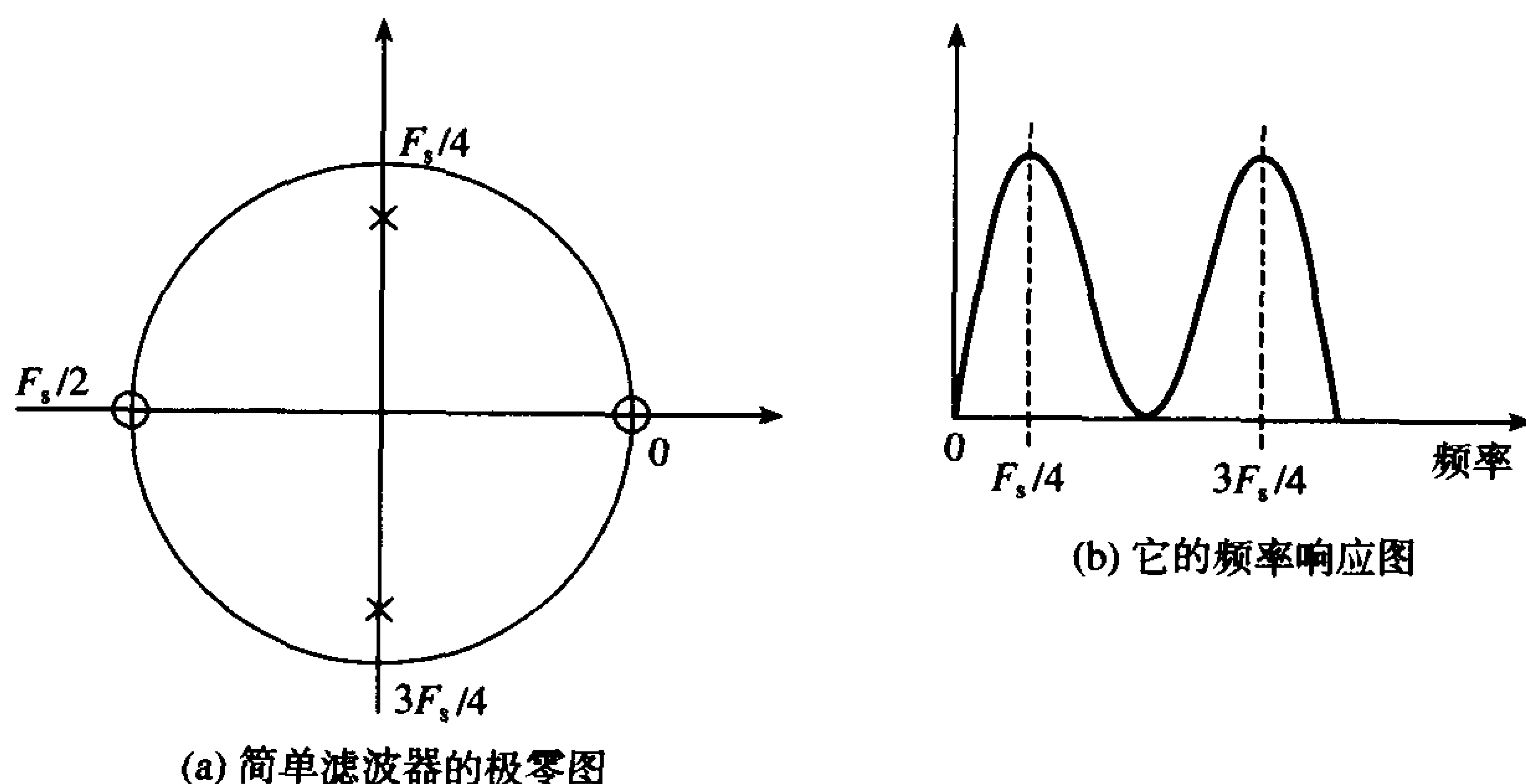
- 极－零点放置法
- 冲激不变法
- 匹配z变换法
- 双线性z变换法

8.5　系数计算的极－零点放置法

8.5.1　基本概念和设计举例

当一个零点放置在z平面一个给定的点上时，频率响应在对应的点为零。另一方面，极点则在对应的频率点产生峰值；参见图 8.3。靠近单位圆的极点产生大的波峰；反之，接近或在单位圆上的零点产生凹槽或者最小值。因此，通过策略性地在z平面上放置极点和零点，我们可以得到简单的低通或者其他的频率选择性滤波器。Lynn and Fuerst(1989)提供了更详细的这种类型滤波器的讨论。

需要记住的重要的一点是：对于实系数的滤波器，极点和零点必须是实的（位于正实轴或负实轴上），或者是复共轭对。我们将用例子来解释这种方法。



(a) 简单滤波器的极零图

(b) 它的频率响应图

图 8.3　简单滤波器的极零图和其频率响应图

例8.1　举例说明计算滤波器系数的简单的极－零点法　一个带通数字滤波器要求满足下面的规范：

(1) 信号在 dc 和 250 Hz 被完全滤去

(2) 中心频率为 125 Hz 的一个窄的通带

(3) 3 dB 带宽为 10 Hz

假设抽样频率是500 Hz，通过在z平面适当地放置极点和零点，得到滤波器的传递函数以及它的差分方程。

解：

首先，我们必须确定在z平面的哪些位置放置极点和零点。因为要求在0和250 Hz完全滤去信号，我们需要在z平面的对应点上放置零点，也就是在单位圆上的0°角和360° × 250/500 = 180°角处放置零点。为了得到中心频率是125 Hz的通带，要求我们将极点放置在±360° × 125/500 = ±90°。为了确保系数是实的，必须有一个复共轭极点对。

极点的半径r是通过期望的带宽求得的。当$r > 0.9$时，r和带宽bw间的近似关系由下式给定：

$$r \simeq 1 - (\mathrm{bw}/F_s)\pi \tag{8.5}$$

本题中，bw = 10 Hz，F_s = 500 Hz，那么可得$r = 1-(10/500)\pi = 0.937$。极零图如图8.4(a)所示。观察极零图可以写出如下的传递函数：

$$H(z) = \frac{(z-1)(z+1)}{(z - r\mathrm{e}^{\mathrm{j}\pi/2})(z - r\mathrm{e}^{-\mathrm{j}\pi/2})}$$

$$= \frac{z^2 - 1}{z^2 + 0.877\,969} = \frac{1 - z^{-2}}{1 + 0.877\,969 z^{-2}}$$

差分方程为

$$y(n) = -0.877\,969 y(n-2) + x(n) - x(n-2)$$

将传递函数$H(z)$和通常的IIR方程（8.2式）进行比较，我们发现这个滤波器是二阶的，它的系数为

$$b_0 = 1 \qquad a_1 = 0$$

$$b_1 = 0 \qquad a_2 = 0.877\,969$$

$$b_2 = -1$$

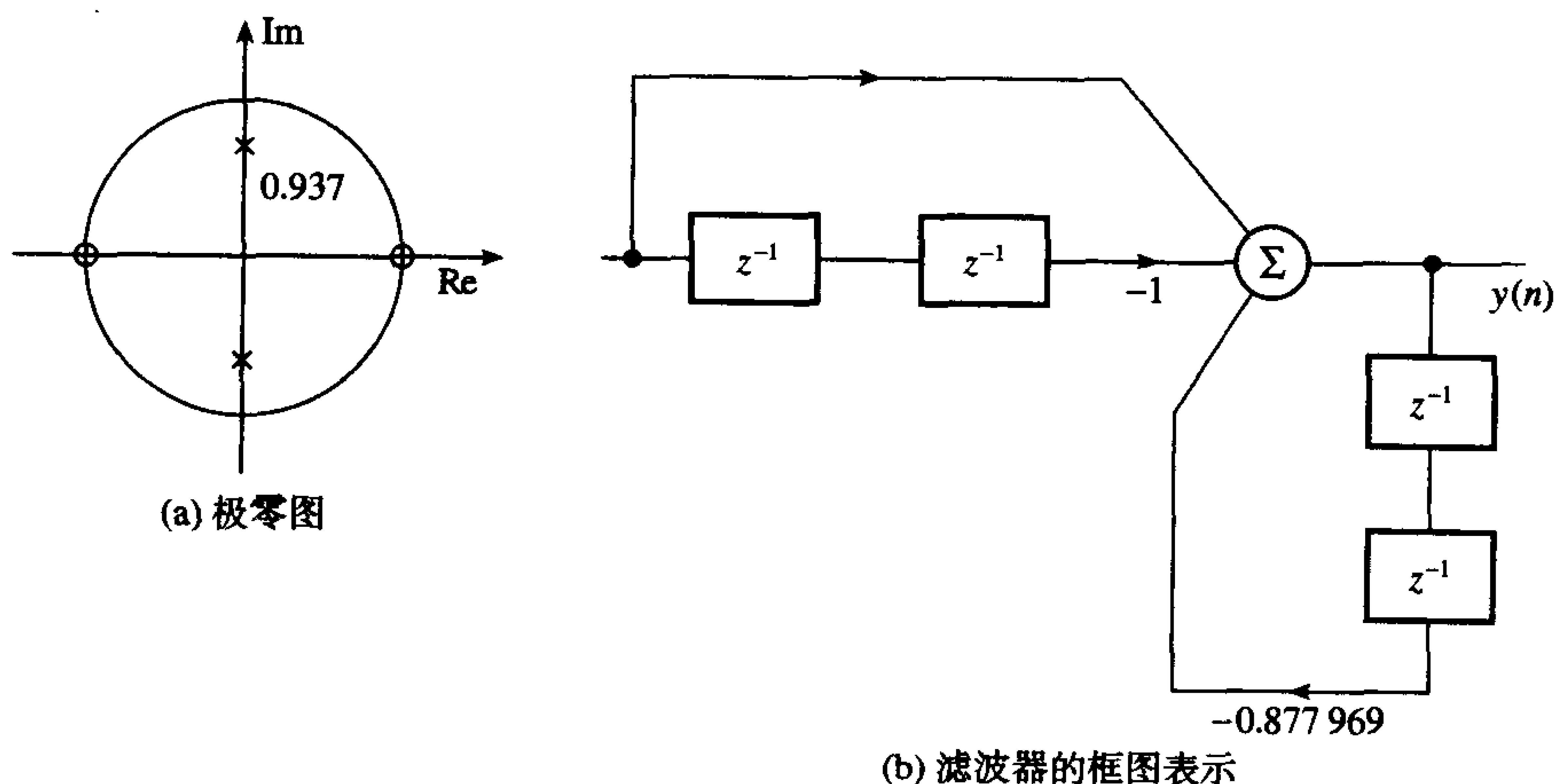


图8.4　极零图和滤波器的框图表示

例8.2　利用极－零点放置法来计算一个陷波滤波器的系数　通过极－零点放置法，求一个简单的陷波数字滤波器的传递函数和差分方程，这个滤波器满足以下规范：

陷波频率	50 Hz
陷波的 3 dB 带宽	± 5 Hz
抽样频率	500 Hz

解：

- 为了滤去50 Hz的分量，我们在单位圆上对应50 Hz的位置放置相应的复零点对，就是在360° × 50/500 = ± 36° 处。
- 为了得到一个尖锐的陷波滤波器，并且改善陷波频率两边的幅度响应，一对复共轭极点放置在半径 $r < 1$ 处。陷波的宽度是由极点的位置决定的。可利用例 8.1 提供的带宽和半径的关系式。这样得到的极点的半径是 0.9372。
- 极零图如图 8.5(a)所示。从图中可得滤波器的传递函数为

$$H(z) = \frac{[z - \exp(-j36°)][z - \exp(j36°)]}{[z - 0.937\exp(-36°)][z - 0.9372\exp(36°)]}$$

$$= \frac{z^2 - 1.6180z + 1}{z^2 - 1.5164z + 0.8783} = \frac{1 - 1.6180z^{-1} + z^{-2}}{1 - 1.5164z^{-1} + 0.8783z^{-2}}$$

差分方程为

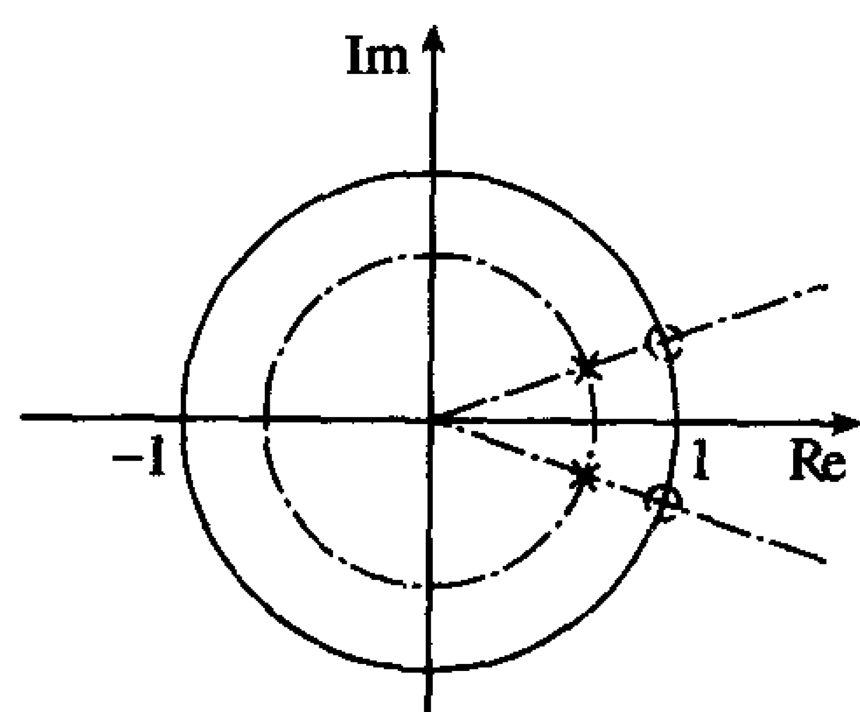
$$y(n) = x(n) - 1.6180x(n-1) + x(n-2) + 1.5164y(n-1) - 0.8783y(n-2)$$

把 $H(z)$和 8.2 式进行比较，可以看出陷波滤波器的系数为

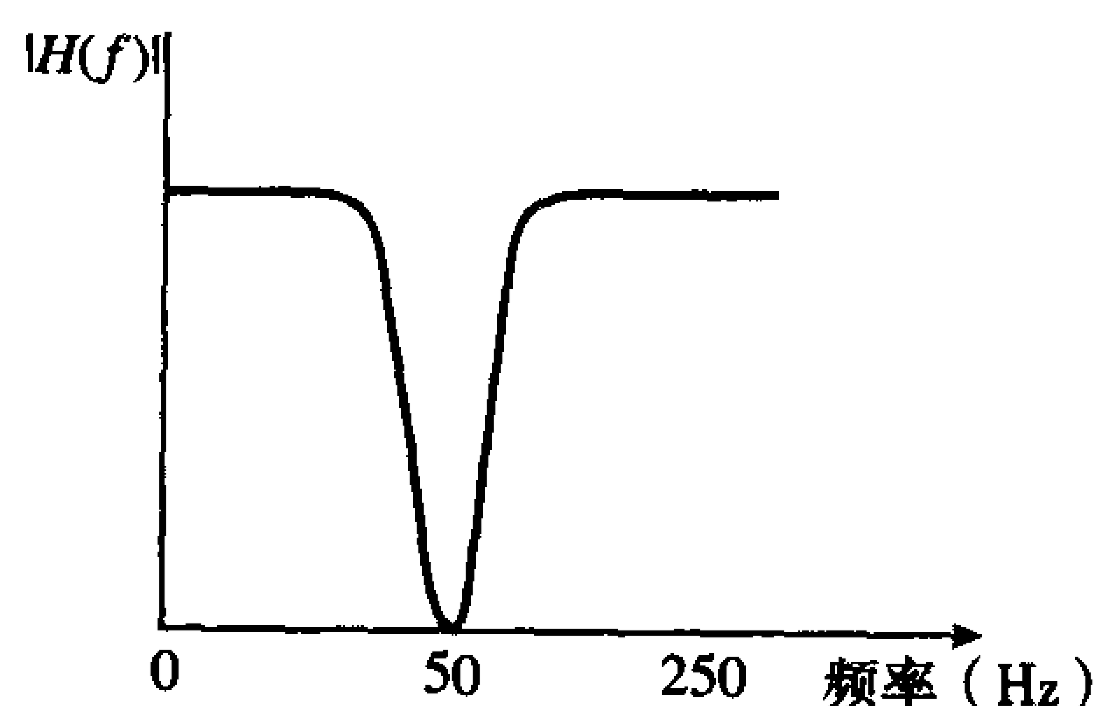
$$b_0 = 1 \qquad a_1 = -1.5164$$

$$b_1 = -1.6180 \qquad a_2 = 0.8783$$

$$b_2 = 1$$



(a) 例8.2的极零图



(b) 对应的频率响应

图 8.5　极零图和对应的频率响应

8.6　系数计算的冲激不变法

8.6.1　基本概念和设计举例

在这种方法中，我们从合适的模拟传递函数 $H(s)$开始入手，利用拉普拉斯变换得到冲激响应 $h(t)$。这样得到的 $h(t)$通过适当的抽样得到 $h(nT)$，且通过对 $h(nT)$进行 z 变换得到期望的传递函数 $H(z)$，其中 T 是抽样间隔。我们将通过例子来说明这种方法。

例 8.3　举例说明冲激不变法　利用冲激不变法进行数字化，简单的模拟滤波器的传递函数为

$$H(s) = \frac{C}{s - p} \tag{8.6}$$

解：

通过拉普拉斯反变换给出冲激响应 $h(t)$：

$$h(t) = L^{-1}[H(s)] = L^{-1}\left(\frac{C}{s-p}\right) = Ce^{pt}$$

其中 L^{-1} 表示拉普拉斯反变换的符号。根据冲激不变法，等价的数字滤波器的冲激响应 $h(nT)$ 等于离散时间 $t=nT$（$n=0, 1, 2,\ldots$）时的 $h(t)$，即

$$h(nT) = h(t)\big|_{t=nT} = Ce^{pnT}$$

传递函数 $H(z)$ 是通过对 $h(nT)$ 进行 z 变换得到的：

$$H(z) = \sum_{n=0}^{\infty} h(nT)z^{-n} = \sum_{n=0}^{\infty} Ce^{pnT}z^{-1}$$

$$= \frac{C}{1-e^{pT}z^{-1}}$$

因此，从上面的结果我们可以写出

$$\frac{C}{s-p} \to \frac{C}{1-e^{pT}z^{-1}} \tag{8.7}$$

为了将冲激不变法应用到具有简单极点的高阶 IIR 滤波器（例如 M 阶），首先利用部分分式将传递函数 $H(s)$ 展开成单极点滤波器之和：

$$H(s) = \frac{C_1}{s-p_1} + \frac{C_2}{s-p_2} + \ldots + \frac{C_M}{s-p_M}$$

$$= \sum_{K=1}^{M} \frac{C_K}{s-p_K} \tag{8.8}$$

其中 p_K 是 $H(s)$ 的极点。8.8 式右边的每一项都有和 8.6 式相同的形式，这样可以利用 8.8 式给出的变换。因此：

$$\sum_{K=1}^{M} \frac{C_K}{s-p_K} \to \sum_{K=1}^{M} \frac{C_K}{1-e^{p_K T}z^{-1}} \tag{8.9}$$

高阶 IIR 滤波器通常用标准的二阶滤波器串联或并行组合来实现。因此，$M=2$ 是我们特别感兴趣的情况。在这种情形下，8.9 式的传递函数变成

$$\frac{C_1}{s-p_1} + \frac{C_2}{s-p_2} \to \frac{C_1}{1-e^{p_1T}z^{-1}} + \frac{C_2}{1-e^{p_2T}z^{-1}}$$

$$= \frac{C_1 + C_2 - (C_1e^{p_2T} + C_2e^{p_1T})z^{-1}}{1-(e^{p_1T}+e^{p_2T})z^{-1} + e^{(p_1+p_2)T}z^{-2}} \tag{8.10}$$

如果极点 p_1 和 p_2 是复共轭的，那么 C_1 和 C_2 也是复共轭的，8.10 式可化简为

$$\frac{C_1}{1-e^{p_1T}z^{-1}} + \frac{C_1^*}{1-e^{p_1^*T}z^{-1}} = \frac{2C_r - [C_r\cos(p_iT) + C_i\sin(p_iT)]2e^{p_rT}z^{-1}}{1-2e^{p_rT}\cos(p_iT)z^{-1} + e^{2p_rT}z^{-2}} \tag{8.11}$$

其中 C_r 和 C_i 是 C_1 的实部和虚部，p_r 和 p_i 是 p_1 的实部和虚部，* 表示复共轭。

对于大多数实际的冲激不变 IIR 滤波器，8.7 式、8.10 式或 8.11 式给出的变换是求传递函数的系数所要求的惟一变换。在附录中给出了一个计算冲激不变滤波器的系数的 C 语言程序，我们将通过一个例子来说明这种转换。

例8.4　应用冲激不变法来设计滤波器　要求设计一个数字滤波器，使它近似等于下面归一化的模拟传递函数：

$$H(s)=\frac{1}{s^2+\sqrt{2}s+1}$$

使用冲激不变法求数字滤波器的传递函数，假设 3 dB 的截止频率 $H(z)$是 150 Hz，抽样频率是 1.28 kHz。

解：

在应用冲激不变法之前，我们需要对频率进行伸缩（scale）来归一化传递函数。这是通过用 s/α代替 s 来实现的，其中 $\alpha=2\pi\times150=942.4778$，以便确保得到的滤波器具有期望的响应，因此

$$H'(s)=H(s)\big|_{s=s/\alpha}=\frac{\alpha^2}{s^2+\sqrt{2}\,\alpha s+\alpha^2}=\frac{C_1}{s-p_1}+\frac{C_2}{s-p_2}$$

其中

$$p_1=\frac{-\sqrt{2}\,\alpha(1-\mathrm{j})}{2}=-666.4324(1-\mathrm{j}),\ p_2=p_1^*$$

$$C_1=-\frac{\alpha}{\sqrt{2}}\mathrm{j}=-666.4324\mathrm{j};\ C_2=C_1^*$$

由于极点是复共轭的，利用 8.11 式中的变换来得到离散时间传递函数$H(z)$。对于本题，$C_r=0$, $C_i=-666.4324$, $p_iT=0.5207$, $p_rT=-0.5207$, $e^{p_rT}=0.5941$, $\sin(p_iT)=0.4974$, $\cos(p_iT)=0.8675$和 $e^{p_rT}=0.3530$。把这些值代入进 8.11 式，我们得到 $H(z)$：

$$H(z)=\frac{393.9264z^{-1}}{1-1.0308z^{-1}+0.3530z^{-2}}$$

如果我们在上面的等式中代入 $z=e^{j\omega T}$，那么在 $\omega=0$ 时 $H(z)$的值是 1223，近似地等于抽样频率。这样大的增益是冲激不变滤波器的特征。一般来说，通过这种方法得到的传递函数的增益等于抽样频率，也就是 $1/T$，这是由于对冲激响应进行抽样而得来的。为了使增益降低，使得滤波器在实现时避免溢出，通常在实践中把 $H(z)$乘以 T（或者等价的让它除以抽样频率）。因此，对于本题，传递函数变为

$$H(z)=\frac{0.3078z^{-1}}{1-1.0308z^{-1}+0.3530z^{-2}}$$

这样我们有

$$b_0=0 \qquad a_1=-1.0308$$
$$b_1=0.3078 \qquad a_2=0.3530$$

消除抽样频率对滤波器增益影响的另一种方法是利用归一化频率。因此在后面的例子里，我们将利用 $T=1$ 和 $\alpha=2\pi\times150/1280=0.7363$。在 8.11 式中利用这些值可以直接推导出上面期望的传递函数。采用归一化频率的一个重要优势是涉及的数字要简单得多。它通常意味着结论可以推广。图 8.6 给出了这个滤波器的框图。

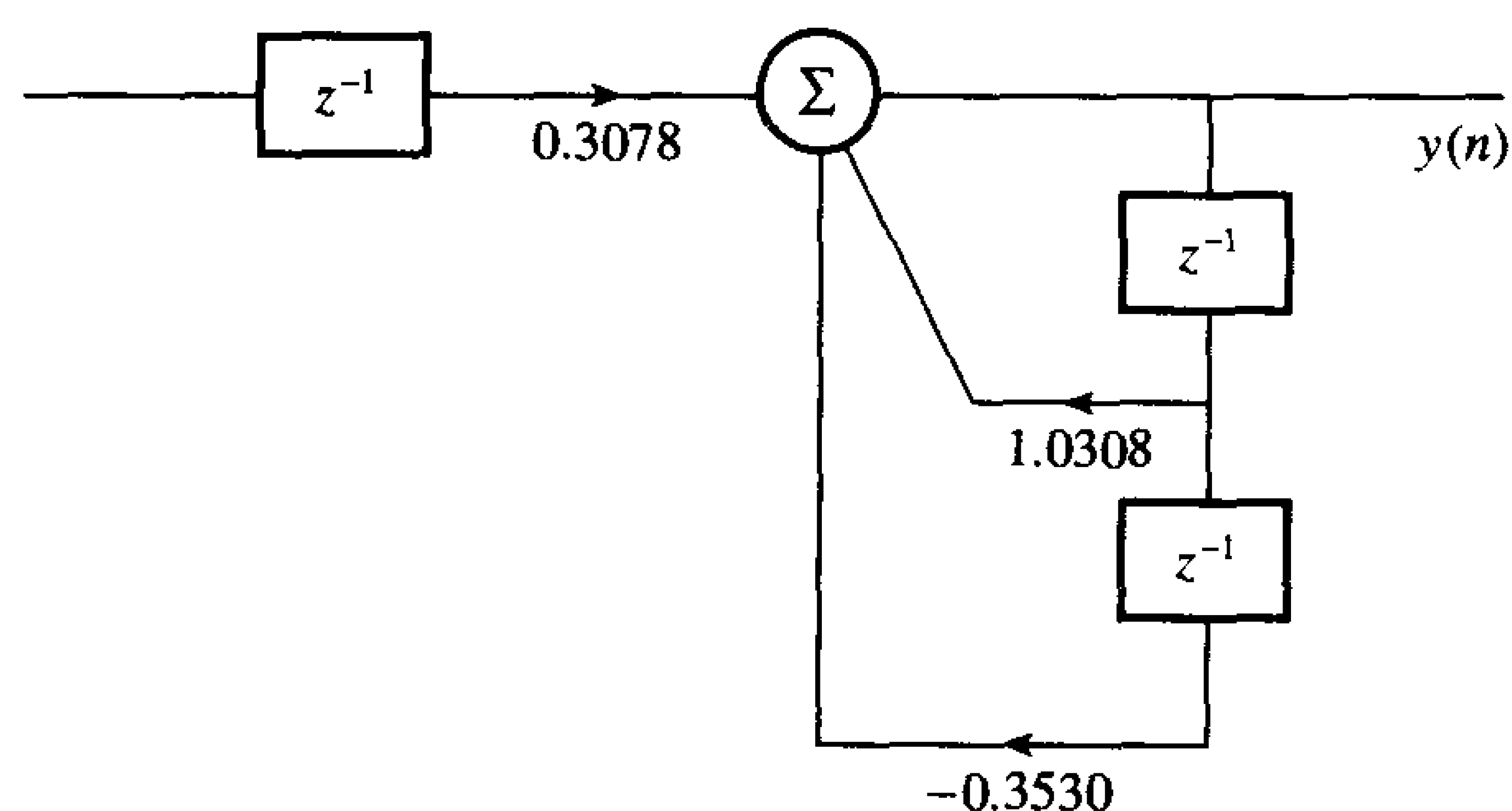


图 8.6　例 8.4 中的滤波器的框图表示

8.6.2　冲激不变法小结

(1) 确定归一化的模拟滤波器 $H(s)$，使它满足期望的数字滤波器的性能规范。

(2) 如果需要，利用部分分式展开 $H(s)$，以便简化下一个步骤。

(3) 求每一部分分式的 z 变换来求 8.9 式。

(4) 把部分分式的 z 变换合并为二阶项以及可能的一阶项，得到 $H(z)$。如果用到实际的抽样频率，那么将 $H(z)$乘以 T。

8.6.3　冲激不变法的注释

(1) 离散滤波器的冲激响应 $h(nT)$，等于模拟滤波器 $h(t)$在离散时刻 $t = nT, n = 0, 1,...$的值；例如图 8.7 所示。因为这个原因，该方法称为冲激不变法。

(2) 抽样频率影响冲激不变离散滤波器的频率响应。为了使频率响应接近等价的模拟滤波器的频率响应，一个足够高的抽样频率是必需的。

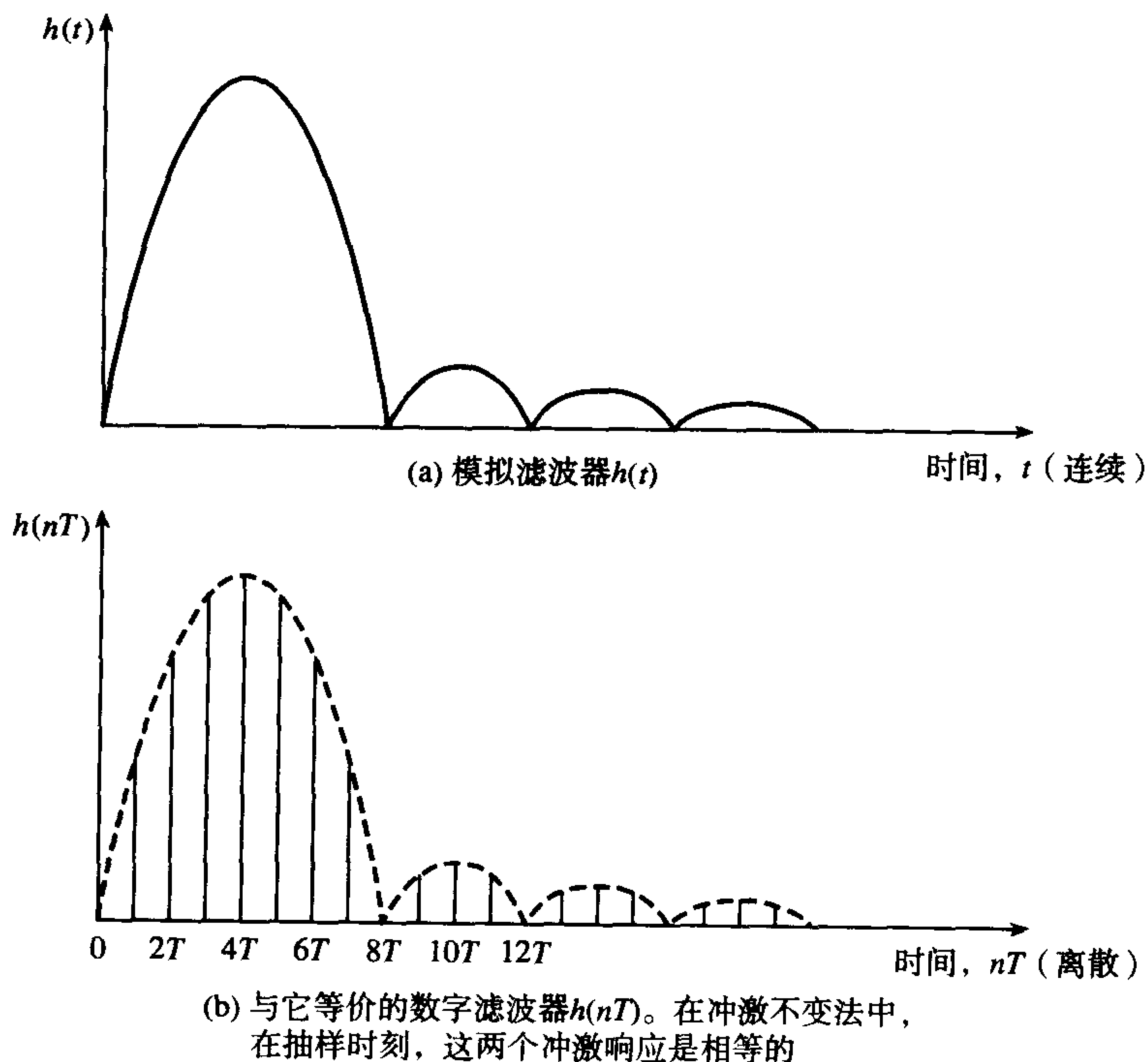


图 8.7　冲激响应的比较

(3) 对一个抽样的数据系统来说，对应于$H(z)$的冲激不变滤波器的频谱将等于原始的模拟滤波器$H(s)$的频谱，但是如图8.8所示的那样，频谱按抽样频率的倍数重复，这样会引起频谱的混叠。然而，如果原始的模拟滤波器的下滑足够陡峭，或者如果模拟滤波器在应用冲激不变法之前是带限的，则混叠会比较微弱。提高抽样频率也可减少混叠。我们可以得出这样的结论：假若抽样频率适当地提高，那么这个方法可以用于几乎无混叠的锐截止的低通滤波器。不过对于高通或者带阻滤波器这个方法并不合适，除非采用抗混叠滤波器。

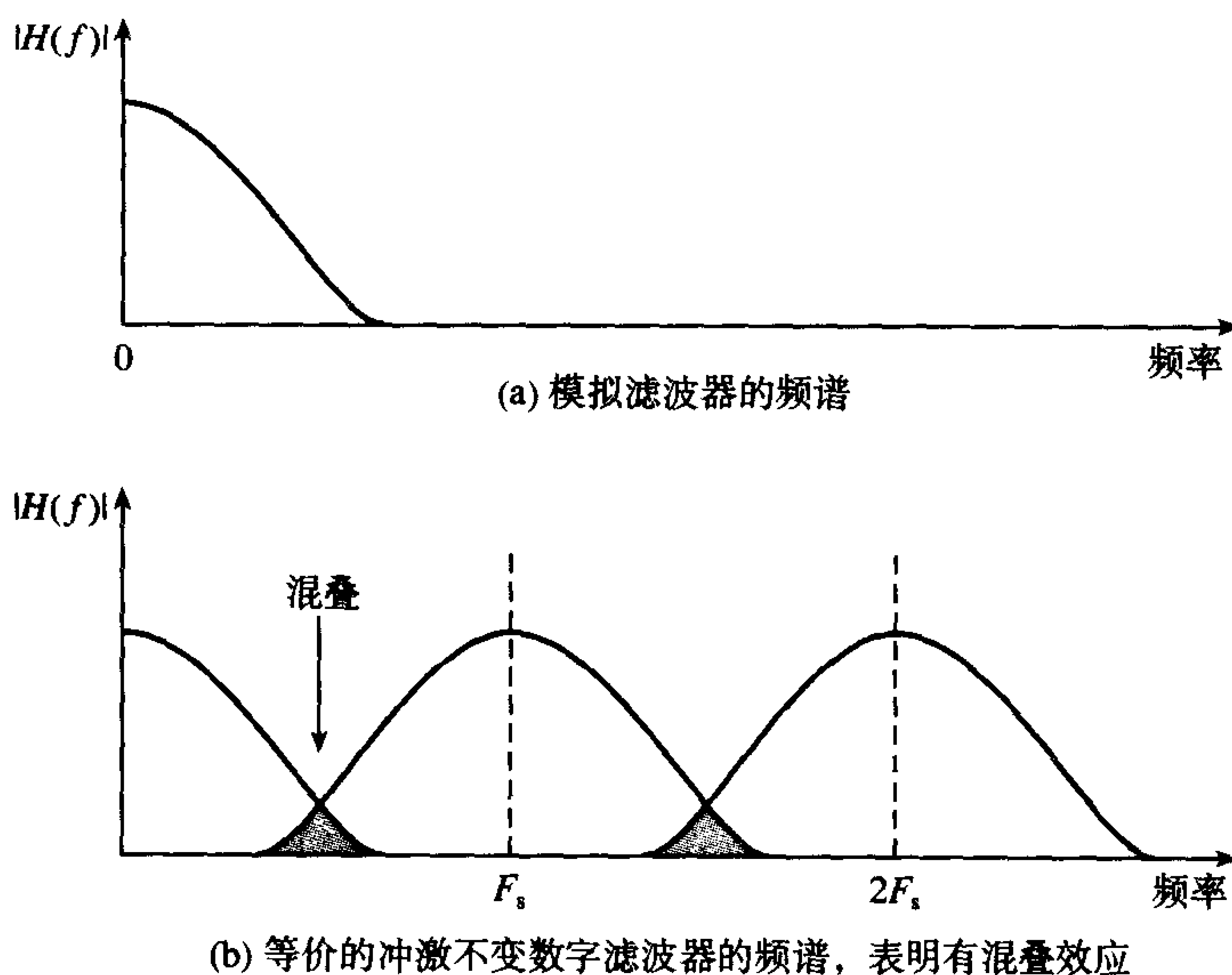


图 8.8　两种等价的滤波器的比较

8.7　系数计算的匹配 z 变换（MZT）法

8.7.1　基本概念和设计举例

匹配z变换法为模拟滤波器转换为等价的数字滤波器提供了一个简单的途径。在MZT法中，模拟滤波器的每一个极点和零点，可以利用下面的式子直接从s平面映射到z平面：

$$(s-a)\to(1-z^{-1}e^{aT}) \tag{8.12}$$

其中T是抽样周期。8.12式把s平面的$s=a$位置的极点（或零点）映射到z平面的$z=e^{aT}$的极点（或零点）。

对于高阶模拟滤波器，传递函数有几个极点和（或者）零点，它们都需要从s平面映射到z平面。对一个具有不同极点和零点的高阶模拟滤波器，传递函数可以写成如下形式：

$$H(s)=\frac{(s-z_1)(s-z_2)\ldots(s-z_M)}{(s-p_1)(s-p_2)\ldots(s-p_N)} \tag{8.13}$$

其中z_k和p_k分别是$H(s)$的零点和极点。

MZT 可以分别应用到每一个因式：

$$(s-z_k)\to(1-z^{-1}e^{z_kT})$$

$$(s-p_k)\to(1-z^{-1}e^{p_kT})$$

在高阶IIR滤波器中，二阶滤波器单元是基本的构建块。因此，8.13式当$M=N=2$时的情形是我们特别感兴趣的。在这种情形下，模拟传递函数化简为

$$H(s)=\frac{(s-z_1)(s-z_2)}{(s-p_1)(s-p_2)} \tag{8.14}$$

对这个式子应用MZT可以给出：

$$\frac{(s-z_1)(s-z_2)}{(s-p_1)(s-p_2)} \to \frac{1-(e^{z_1T}+e^{z_2T})z^{-1}+e^{(z_1+z_2)T}z^{-2}}{1-(e^{p_1T}+e^{p_2T})z^{-1}+e^{(p_1+p_2)T}z^{-2}} \tag{8.15}$$

如果二阶部分的极点和零点是以复共轭对形式出现，那么$p_2=p_1^*$和$z_2=z_1^*$，8.15式的右边可以简化为

$$\frac{1-2e^{z_rT}\cos(z_iT)z^{-1}+e^{z_rT}z^{-2}}{1-2e^{p_rT}\cos(p_iT)z^{-1}+e^{p_rT}z^{-2}} \tag{8.16}$$

其中z_r和z_i、p_r和p_i分别是z_1和p_1的实数和虚数部分。

实际上，把二阶模拟滤波器部分表示为熟悉的有理多项式的形式更为方便：

$$H(s)=\frac{(s-z_1)(s-z_2)}{(s-p_1)(s-p_2)}=\frac{A_0+A_1s+A_2s^2}{B_0+B_1s+B_2s^2}$$

那么$H(s)$的极点和零点为

$$p_{1,2}=-\frac{B_1}{2B_2}\pm\left[\left(\frac{B_1}{2B_2}\right)^2-\frac{B_0}{B_2}\right]^{\frac{1}{2}} \tag{8.17a}$$

$$z_{1,2}=-\frac{A_1}{2A_2}\pm\left[\left(\frac{A_1}{2A_2}\right)^2-\frac{A_0}{A_2}\right]^{\frac{1}{2}} \tag{8.17b}$$

实际上，给定模拟滤波器的传递函数，8.17a式和8.17b式允许我们用简单的方法来求出极点和零点的位置（也就是它们的实部和虚部）。一旦我们知道$H(s)$的零点和极点的实部和虚部，我们就可以利用8.15式或者8.16式求出等价的离散滤波器的传递函数$H(z)$。

例8.5 一个模拟滤波器的归一化传递函数由下式给出：

$$H(s)=\frac{1}{s^2+\sqrt{2}s+1}$$

利用匹配z变换法求等价的数字滤波器的传递函数$H(z)$。假设3 dB的截止频率是150 Hz，抽样频率是1.28 kHz。

解：

截止频率可以表示为$\omega_c=2\pi\times150=942.4778$弧度/秒（rad/s），用$s/\omega_c$代替$s$得到没有归一化的模拟滤波器的传递函数：

$$H'(s)=H(s)\Big|_{s=\frac{s}{\omega_c}}=\frac{\omega_c^2}{s^2+\sqrt{2}\omega_cs+\omega_c^2}$$

滤波器的极点位于

$$p_{1,2} = -\frac{\sqrt{2}\omega_c}{2} \pm \left[\left(\frac{\sqrt{2}\omega_c}{2}\right)^2 - \omega_c^2\right]^{\frac{1}{2}}$$

$$= -\frac{\sqrt{2}\omega_c}{2} \pm \omega_c \left[\left(\frac{\sqrt{2}}{2}\right)^2 - 1\right]^{\frac{1}{2}}$$

$$= -\frac{\sqrt{2}\omega_c}{2}(1 \mp \mathrm{j})$$

对于本题，极点的实部和虚部为

$$p_r = -\frac{\sqrt{2}\omega_c}{2} = -666.4324, \quad p_i = \frac{\sqrt{2}\omega_c}{2}\mathrm{j} = 666.4324\mathrm{j}$$

因此，$p_rT = -0.520\,650\,3$, $p_iT = 0.520\,650\,3$, $\cos(p_iT) = 0.867\,496$，$e^{p_rT} = 0.594\,134$。最后得到的传递函数变为

$$H(z) = \frac{8.8876 \times 10^5}{1 - 1.030\,818z^{-1} + 0.594\,134z^{-2}}$$

8.7.2　匹配 z 变换法小结

(1) 确定一个合适的模拟传递函数 $H(s)$，使它满足期望的数字滤波器的技术规范。

(2) 求出 $H(s)$的极点和零点的位置。这可能要求对模拟传递函数 $H(s)$做因式分解。

(3) 利用 8.12 式把 s 平面的极点和零点映射到 z 平面。对于二阶项，可以用 8.15 式和 8.16 式。

(4) 适当地合并 z 平面的方程以得到传递函数 $H(z)$。

8.7.3　匹配 z 变换法的注释

(1) MZT法要求已知模拟滤波器的极点和零点的位置。这可以通过对传递函数$H(s)$做因式分解得到。因而 MZT 法相对比较容易应用。

(2) MZT 和冲激不变法得出的离散滤波器具有相同的分母。例如，比较 8.15 式的 MZT 的分母和 8.10 式中给出的冲激不变法得到的结果。如果我们比较在例 8.4 和例 8.5 得到的传递函数的分母，这个结论也是很明显的。

(3) 在数字滤波器里，有用的频带是从零扩展到奈奎斯特频率（抽样频率的一半），而在模拟滤波器里它是从零到无限。因而，MZT 映射和其他的映射一样，是将无限的模拟频带压缩到有限的频带。和模拟滤波器相比，这会引起等价的数字滤波器的频率响应的失真。对 MZT 来说，最后得到的滤波器和模拟滤波器相比，趋向于提供比较小的衰减。在 8.12 节中，我们将进一步探讨这个问题并说明如何利用这个特性。

(4) 如果模拟滤波器的极点频率接近奈奎斯特频率或者零点频率超过奈奎斯特频率（即抽样频率的一半），那么最后得到的基于 MZT 的数字滤波器会因为频率混叠而失真（参见后面）。在这种情况下，超过奈奎斯特频率的模拟滤波器的响应仍然是有意义的，通过隐含的抽样过程将其折叠到期望的频带里。

(5) MZT 对数字化一个全极点的模拟滤波器是不合适的，因为在奈奎斯特频率以上缺乏零点。这种情况下问题可以通过在 $z = -1$（也就是在奈奎斯特频率处）增加零点来缓解。

在 8.12 节里将深入讨论 MZT 对频率响应的影响。

8.8 系数计算的双线性 z 变换（BZT）法

8.8.1 基本概念和设计举例

这是到目前为止求IIR滤波器系数的最重要的方法。在BZT方法中，基本的运算是把模拟滤波器 $H(s)$ 转换成一个等价的数字滤波器，它是通过如下式子来代替 s 的：

$$s = k\frac{z-1}{z+1},\ k = 1 \text{ 或 } \frac{2}{T} \tag{8.18a}$$

上面的变换式，也就是如图8.9所示的那样，把模拟传递函数 $H(s)$ 从 s 平面映射到 z 平面的离散传递函数 $H(z)$。我们注意到在图中 s 平面的整个 $j\omega$ 轴都映射到单位圆上，s 平面的左半平面映射到单位圆内，右半平面映射到单位圆外。因此，一个稳定的模拟滤波器，也就是极点都在 s 平面左半平面，将得出一个极点都在单位圆内的数字滤波器。

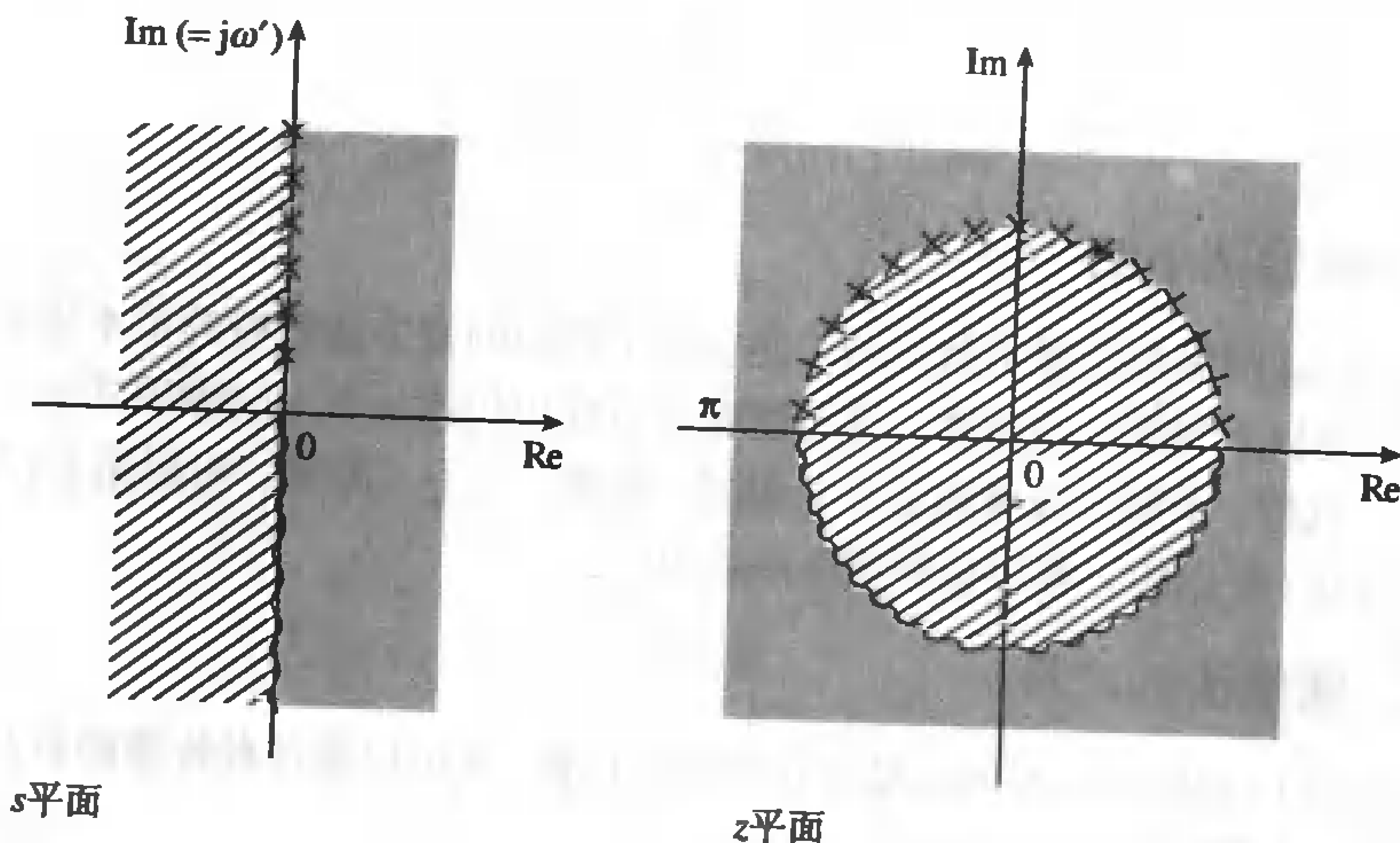


图8.9 利用双线性 z 变换从 s 平面映射到 z 平面说明图。注意 s 平面的正的 $j\omega'$ 轴（也就是 $s=0$ 到 $s=j\infty$）映射到单位圆的上半部分，负的 $j\omega'$ 轴映射到下半部分

遗憾的是，像在8.18a式那样直接在 $H(s)$ 里替换 s，可能得出的数字滤波器会含有不希望的响应。通过在8.18a式中进行替代 $z = e^{j\omega T}$ 和 $s = j\omega'$，很容易证明这一点。简单来说，我们发现模拟频率 ω' 和数字频率 ω 有下面的关系：

$$\omega' = k\tan\left(\frac{\omega T}{2}\right),\ k = 1 \text{ 或 } \frac{2}{T} \tag{8.18b}$$

8.18b式画于图8.10中。可以看出，对于较小的 ω 值，模拟频率 ω' 和数字频率 ω 的关系几乎是线性的；但是对较大的 ω 值，它们的关系就变成非线性了，这就引起了数字频率响应的失真（或者说弯曲）。注意，例如，左边的模拟滤波器的通带是一个恒定的带宽，且以有规律的间隔为中心。然而，等价的数字滤波器的通带却逐步压缩。这种影响一般可以通过让模拟滤波器在应用双线性变换前进行预弯曲来补偿。

为了补偿这种影响，我们在应用BZT之前对一个或多个关键的频率进行预弯曲。例如，对于一个低通滤波器，我们经常如下预弯曲截止频率或带沿频率：

$$\omega_p' = k\tan\left(\frac{\omega_p T}{2}\right) \tag{8.19}$$

其中

ω_p = 指定的截止频率

ω'_p = 预弯曲的截止频率

$k = 1$ 或 $T/2$

T = 抽样周期

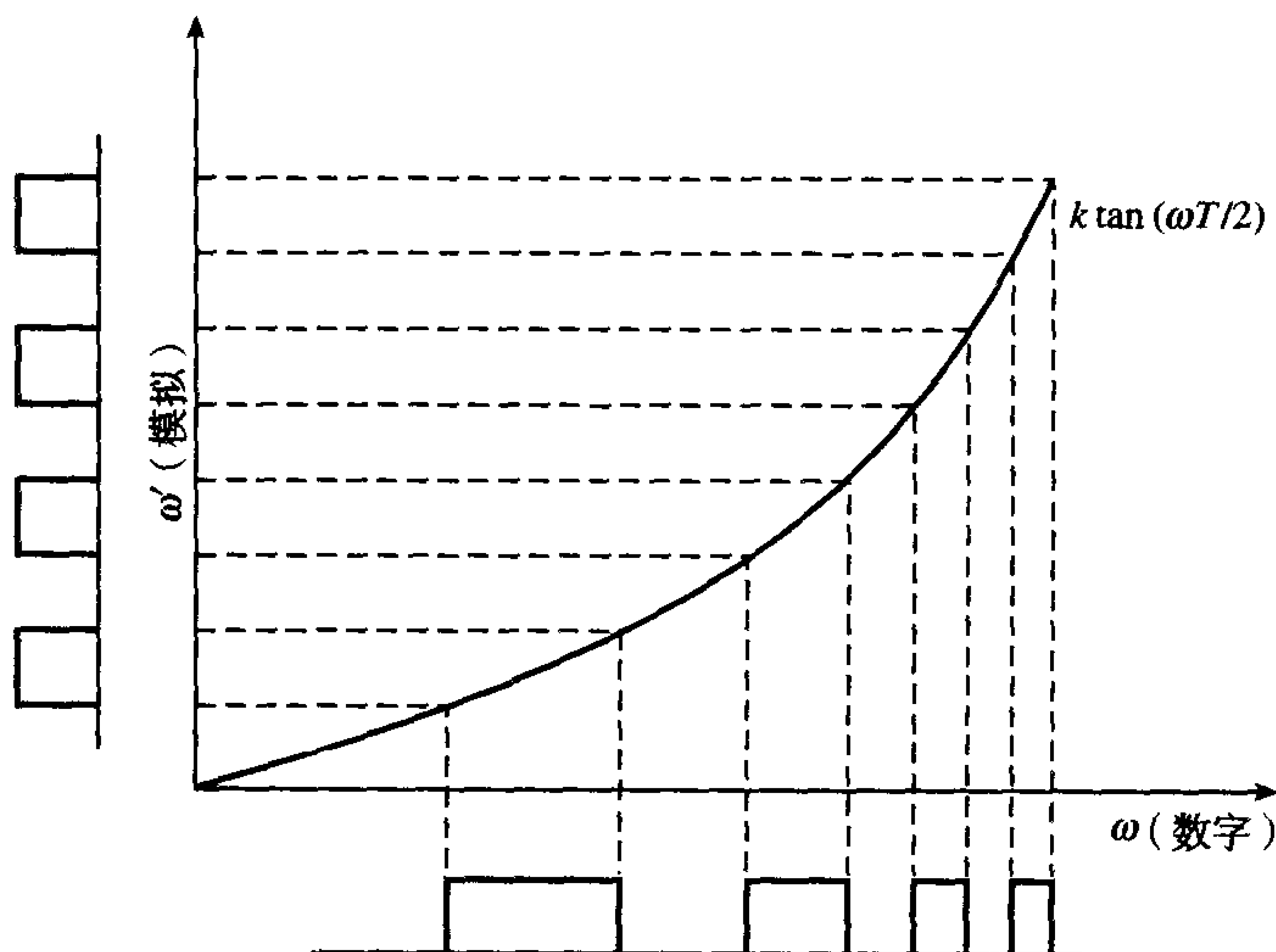


图 8.10　模拟频率和数字频率之间的关系来表明具有弯曲效应。注意，在模拟情况下，等间隔的通带在转换到数字域后，高频段被挤到一起

8.8.2　系数计算的 BZT 法小结

对于标准的频率选择 IIR 滤波器，使用 BZT 方法的步骤可以总结如下：

(1) 利用数字滤波器的规范找一个合适的归一化的模拟低通滤波器的原型 $H(s)$。

(2) 确定期望滤波器的带沿频率和边界频率，并且对它们进行预弯曲。对于低通或者高通滤波器，这正好是一个带沿频率或者截止频率（即 ω_p）。对于带通和带阻滤波器，我们有上带沿频率和下带沿频率 ω_{p1} 和 ω_{p2}，这两个带沿频率都需要预弯曲（阻带边沿频率可能也需要指定）：

$$\omega'_p = \tan\left(\frac{\omega_p T}{2}\right) \tag{8.20a}$$

$$\omega'_{p1} = \tan\left(\frac{\omega_{p1} T}{2}\right);\quad \omega'_{p2} = \tan\left(\frac{\omega_{p2} T}{2}\right) \tag{8.20b}$$

(3) 根据要求的滤波器的不同类型，利用下面的转换式子中的一个来替换传递函数 $H(s)$ 中的 s，使模拟原型滤波器反向归一化（denormalize）：

$$s = \frac{s}{\omega'_p}\quad \text{低通到低通} \tag{8.21a}$$

$$s = \frac{\omega'_p}{s}\quad \text{低通到高通} \tag{8.21b}$$

$$s = \frac{s^2 + \omega_0^2}{Ws}\quad \text{低通到带通} \tag{8.21c}$$

$$s = \frac{Ws}{s^2 + \omega_0^2}\quad \text{低通到带阻} \tag{8.21d}$$

其中

$$\omega_0^2 = \omega'_{p2}\omega'_{p1}, \quad W = \omega'_{p2} - \omega'_{p1}$$

(4) 在频率伸缩（即反向归一化）以后的传递函数$H'(s)$中按下式取代s，

$$s = \frac{z-1}{z+1}$$

应用BZT得到希望的数字滤波器的传递函数$H(z)$。

例8.6 低通滤波器 要求设计一个数字低通滤波器来近似下面的传递函数：

$$H(s) = \frac{1}{s^2 + \sqrt{2}s + 1}$$

利用BZT法求数字滤波器的传递函数$H(z)$，假设3 dB的截止频率是150 Hz，抽样频率是1.28 kHz。

解：

边界频率，$\omega_p = 2\pi \times 150$弧度/秒，$F_s = 1/T = 1.28$ kHz，给出的一个预弯曲的边界频率：

$$\omega'_p = \tan(\omega_p T/2) = 0.3857$$

频率伸缩以后的模拟滤波器由下式给出：

$$H'(s) = H(s)\big|_{s=s/\omega'_p} = \frac{1}{(s/\omega'_p)^2 + \sqrt{2}s/\omega'_p + 1}$$

$$= \frac{\omega'^2_p}{s^2 + \sqrt{2}\omega'_p s + \omega'_p s} = \frac{0.1488}{s^2 + 0.5455s + 0.1488}$$

应用BZT得

$$H(z) = H'(s)\Big|_{s=\frac{z-1}{z+1}} = \frac{0.0878z^2 + 0.1756z + 0.0878}{z^2 - 1.0048z + 0.3561}$$

$$= \frac{0.0878(1 - 2z^{-1} + z^{-2})}{1 - 1.0048z^{-1} + 0.3561z^{-2}}$$

例8.7 高通滤波器 一个简单的低通电阻–电容滤波器的归一化传递函数由下式给出：

$$H(s) = \frac{1}{s+1}$$

从s平面方程出发，利用BZT法确定等价的离散时间的高通滤波器的传递函数。假设抽样频率是150 Hz，截止频率是30 Hz。

解：

数字滤波器的边界频率是$\omega_p = 2\pi \times 30$弧度/秒。预弯曲后的截止频率是$\omega'_p = \tan(\omega_p T/2)$。另外有$T = 1/150$ Hz，$\omega'_p = \tan(\pi/5) = 0.7265$。

利用8.21a式的从低通到高通的转换关系，得到反向归一化的模拟传递函数为

$$H'(s) = H(s)\big|_{s=\omega'_p/s} = \frac{1}{\omega'_p/s + 1} = \frac{s}{s + 0.7265}$$

应用BZT可求得z平面的传递函数为

$$H(z) = H'(s)\big|_{s=(z-1)/(z+1)} = \frac{(z-1)/(z+1)}{(z-1)/(z+1)+0.7265}$$

简化后得

$$H(z) = 0.5792\frac{1-z^{-1}}{1+0.1584z^{-1}}$$

离散时间滤波器的系数为

$$b_0 = 0.5792, \quad a_1 = 0.1584$$
$$b_1 = -0.5792$$

例 8.8　带通滤波器　设计一个具有巴特沃斯特性的离散时间带通滤波器，要求满足下面给出的规范：

通带	200 ~ 300 Hz
抽样频率	2 kHz
滤波器阶数，N	2

利用 BZT 方法求滤波器的系数。

解：

求一阶归一化的模拟低通滤波器（因为对于带通滤波器来说，频带转换（参见8.21c 式）会使滤波器阶数加倍）。因此，

$$H(s) = \frac{1}{s+1}$$

预弯曲的边界频率是

$$\omega'_{p1} = \tan\left(\frac{\omega_{p1}T}{2}\right) = \tan\left(\frac{2\pi\times 200}{2\times 2000}\right) = 0.3249$$

$$\omega'_{p2} = \tan\left(\frac{\omega_{p2}T}{2}\right) = \tan\left(\frac{2\pi\times 300}{2\times 2000}\right) = 0.5095$$

$$\omega_0^2 = \omega'_{p1}\omega'_{p2} = 0.1655$$
$$W = \omega'_{p2} - \omega'_{p1} = 0.1846$$

利用从低通到带通的转换，即 8.21c 式，我们有

$$H'(s) = H(s)\Big|_{s=\frac{s^2+\omega_0^2}{Ws}} = \frac{1}{\dfrac{s^2+\omega_0^2}{Ws}+1}$$

$$= \frac{Ws}{s^2+Ws+\omega_0^2}$$

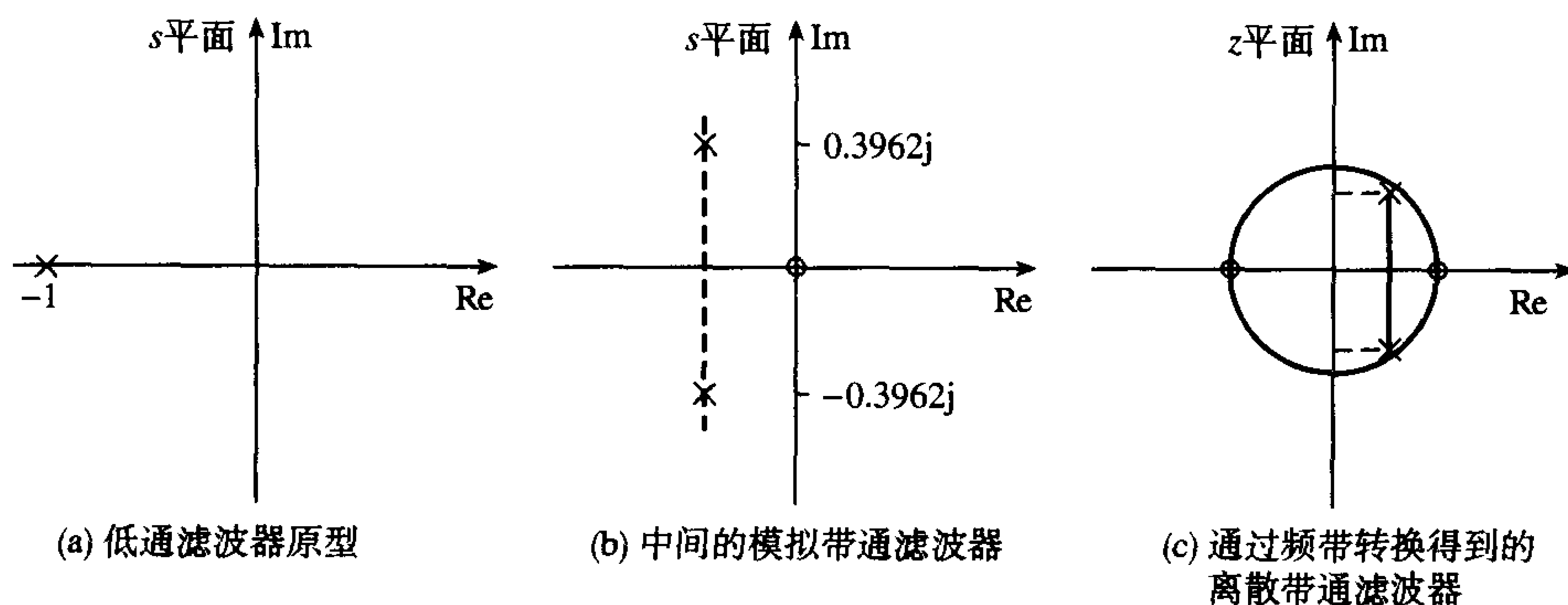
应用 BZT 可得

$$H(z) = H'(s)\Big|_{s=\frac{z-1}{z+1}} = \frac{W\left(\dfrac{z-1}{z+1}\right)}{\left(\dfrac{z-1}{z+1}\right)^2 + W\left(\dfrac{z-1}{z+1}\right) + \omega_0^2}$$

代入ω_0^2和W的值并化简，我们有

$$H(z) = 0.1367\frac{1 - z^{-2}}{1 - 1.2362z^{-1} + 0.7265z^{-2}}$$

归一化的原型低通滤波器（LPF）、模拟带通滤波器和离散时间的带通滤波器的极零图在图8.11中进行了描述。注意，从低通到带通的转换使得在s平面的原点和无穷远处产生了零点，那么，BZT转换把零点映射到$z = \pm 1$。因此，离散带通滤波器的零点是在$z = 1$和$z = -1$。它的极点是$z = 0.6040 \pm 0.6015\text{j}$。模拟带通滤波器的零点是$s = 0$以及无穷远处（没有画出），极点在$s = -0.0923 \pm 0.3962\text{j}$。



(a) 低通滤波器原型　(b) 中间的模拟带通滤波器　(c) 通过频带转换得到的离散带通滤波器

图8.11 极零图

实际应用中，高阶IIR滤波器（例如$N > 3$）通常用二阶和/或一阶部分串联或者并联组合在一起，这减少了有限字长对滤波器性能的影响（参见后面）。因此，在把一个模拟滤波器转换成最终的离散形式的z变换函数$H(z)$后，如果是高阶的，那么需要表示成因式形式（对串联实现结构），或者是二阶和/或一阶部分项之和的形式（对于并联实现结构）。为了简化这个任务，我们可以在开始时把$H(s)$表示成因式分解形式，接着把每一个因式独立转换。最后得到的因式合并在一起或者重新组织成一种合适的离散实现结构形式，从而变成一个适宜于期望的实现形式的格式。这是在例8.15里用到的基本方法。

8.8.3 对双线性变换法的一些说明

实质上，BZT方法包含两个独立的变换。首先，归一化模拟传递函数是通过下式代入s来进行频率伸缩的：

$$s = \frac{s}{\omega_p'} \qquad (8.22a)$$

其中

$$\omega_p' = k\tan\left(\frac{\omega_p T}{2}\right), \quad k = 1 \text{ 或 } 2/T$$

其次，在新的传递函数中按下式代入s来应用BZT：

$$s = k\frac{z - 1}{z + 1} \qquad (8.22b)$$

(1) 在许多教材中（例如，Rabiner and Gold, 1975），在上面的两个操作中利用因子$k = 2/T$是很普遍的做法。应该提醒注意的是，$k = 1$和$k = 2/T$都会推出相同的结果，因为k最后都会被约掉。为了说明这个原因，考虑下面简单的滤波器：

$$H(s) = \frac{1}{s+1}$$

假设数字滤波器要有截止频率 ω_p，那么我们必须用下面的频率对 $H(s)$进行频率伸缩变换：

$$\omega_p' = k\tan\left(\frac{\omega_p T}{2}\right)$$

因此，传递函数为

$$H'(s) = H(s)\big|_{s=s/\omega_p'} = \frac{1}{s/k\tan(\omega_p T/2)+1}$$

接下来，我们用 $k(z-1)/(z+1)$来代替 s：

$$H(z) = H'(s)\big|_{s=k(z-1)/(z+1)} = \frac{1}{[\not{k}(z-1)/(z+1)]/\not{k}\tan(\omega_p T/2)+1}$$

从上式我们可以看出，因子 k 被消掉了，它和 k 是 1 或者 $2/T$ 没有关系。

(2) 从计算效率角度来说，两个传递函数可以合并成一个：

$$s = \cot\left(\frac{\omega_p T}{2}\right)\frac{z-1}{z+1} \tag{8.23}$$

例 8.15 解释了这个方法。

(3) 对于低通和高通滤波器来说，$H(z)$的阶数和 $H(s)$的阶数是相同的。例如，如果 $H(z)$是从一个二阶模拟滤波器 $H(s)$推导出来的，那么 $H(z)$也将是一个二阶系统。对于带通和带阻滤波器，$H(z)$的阶数是 $H(s)$阶数的二倍。这个关系在 BZT 方法里有时用来减少代数运算（参见 Stanley et al., 1984）。

(4) 在实践中，有时我们会遇到这样的情况：需要把一个已经存在的模拟滤波器的 s 平面的传递函数，转化成一个等价的离散时间的带通滤波器。例如，在数字音频里，已经很成功地应用于均衡的模拟滤波器，可能需要转换成一个等价的数字滤波器（Clark et al., 2000）。

在这种情况下，实际滤波器的模拟传递函数已经有了，所以在预弯曲和利用直接的低通到低通的频率伸缩变换以后，BZT 方法就可以直接应用。下面的例子将对所包含的设计问题进行解释。

例 8.9　当实际滤波器的模拟传递函数已经存在时，应用 BZT 方法的一个设计举例　利用 BZT 技术，求出离散时间的钟形（Bell）滤波器的系数。该滤波器用于数字混频器中的音频信号处理，它的控制设置对应于 Q 因子 2，5 kHz 处的 6.02 dB 的提升（boost，峰值）。假设抽样频率为 48 kHz，等价的模拟滤波器的 s 平面的传递函数为

$$H(s) = \frac{s^2 + (3+k)\dfrac{\omega_0}{Q}s + \omega_0^2}{s^2 + (3-k)\dfrac{\omega_0}{Q}s + \omega_0^2}$$

其中

$$k = 3\left(\frac{G-1}{G+1}\right)$$

ω_0 = 提升频率

G = 增益因子

Q = Q 因子

解：

增益或者 6.02 dB 的提升对应于

$$G = 10^{\frac{6.02}{20}} = 1.9999; \quad k = 0.9999$$

s 平面传递函数变为

$$H(s) = \frac{s^2 + 4\dfrac{\omega_0}{Q}s + \omega_0^2}{s^2 + 2\dfrac{\omega_0}{Q}s + \omega_0^2}$$

在这个等式中代入 $s = \mathrm{j}\omega$ 可得到模拟频率响应。

现在预弯曲的提升频率是

$$\omega_0' = \tan\left(\frac{\omega_0 T}{2}\right) = 0.339\,454$$

频率伸缩因子（Clark et al., 2000）如下式给定：

$$p = \frac{\omega_0}{\tan\left(\omega_0 T\big/2\right)} = \frac{\omega_0}{0.339\,454}$$

因此，预弯曲的模拟传递函数是

$$H'(s) = H(s)\big|_{s=ps} = \frac{p^2s^2 + 4\dfrac{\omega_0}{Q}ps + \omega_0^2}{p^2s^2 + 2\dfrac{\omega_0}{Q}ps + \omega_0^2}$$

应用 BZT，

$$H(z) = H'(s)\Big|_{s=\frac{z-1}{z+1}}$$

$$= \frac{\left(\dfrac{z-1}{z+1}\right)^2 + 0.678\,908\,5\left(\dfrac{z-1}{z+1}\right) + 0.115\,229}{\left(\dfrac{z-1}{z+1}\right)^2 + 0.339\,454\left(\dfrac{z-1}{z+1}\right) + 0.115\,229}$$

$$= \frac{1.233\,352 - 1.216\,444z^{-1} + 0.299\,94z^{-2}}{1 - 1.216\,444z^{-1} + 0.533\,294\,6z^{-2}}$$

例 8.10 一个简单的 LRC 陷波滤波器的 s 平面的归一化传递函数如下：

$$H(s) = \frac{s^2 + 1}{s^2 + s + 1}$$

利用 BZT 法，求出等价的离散时间滤波器的传递函数。假设陷波频率为 50 Hz，抽样频率为 500 Hz。

解：

因为我们已经有了 s 平面的传递函数，那么再应用从低通到带阻的转换公式就不合适了，因为这相当于一个双变换。边界频率是

$$\omega_{\mathrm{p}}' = \tan\left(\frac{\omega_{\mathrm{p}} T}{2}\right) = \tan\left(\frac{2\pi \times 50}{500 \times 2}\right) = 0.324\,919\,6$$

频率归一化后的 s 平面传递函数是

$$H'(s) = H(s)\Big|_{s \to \frac{s}{\omega_p}}$$

$$= \frac{\left(\frac{s}{\omega'_p}\right)^2 + 1}{\left(\frac{s}{\omega'_p}\right)^2 + \frac{s}{\omega'_p} + 1}$$

$$= \frac{s^2 + 0.105\,572}{s^2 + 0.324\,919\,6s + 0.105\,572}$$

应用 BZT：

$$H(z) = H'(s)\Big|_{s \to \frac{z-1}{z+1}}$$

$$= \frac{\left(\frac{z-1}{z+1}\right)^2 + 0.105\,572}{\left(\frac{z-1}{z+1}\right)^2 + 0.324\,919\,6\left(\frac{z-1}{z+1}\right) + 0.105\,572}$$

8.9　利用 BZT 和经典的模拟滤波器来设计 IIR 滤波器

在实际应用中，模拟传递函数$H(s)$（从它可以得到$H(z)$）可能并不是已知的，我们必须从期望的数字滤波器的技术规范来确定。对于标准的频率选择性数字滤波任务（比如包括低通、高通、带通和带阻滤波器）来说，$H(s)$可以从具有巴特沃斯、切比雪夫以及椭圆特性质的经典滤波器推导出（参见图 8.12）。在本节里，我们将给出从这些经典模拟滤波器出发来设计 IIR 滤波器的方法。我们将详细考虑基本概念，并用验证过的例子来解释设计中的一些问题。首先，简要回顾一下要设计的 IIR 滤波器的三个经典模拟滤波器的有关特性。仅考虑低通滤波器原型，因为其他的两种滤波器类型通常可以从归一化的低通滤波器推导出，下面将证明这一点。

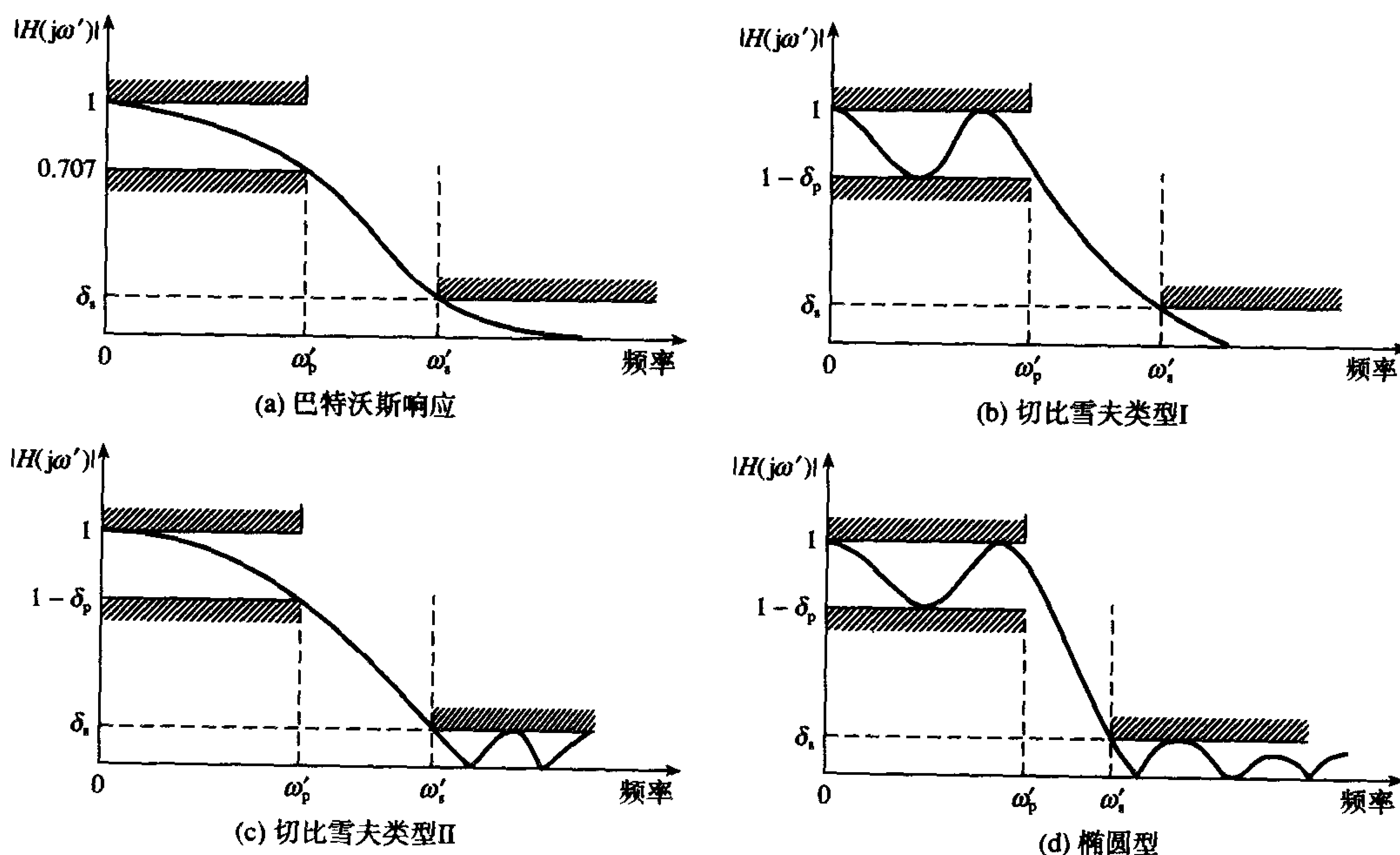


图 8.12　几个经典模拟滤波器的频率响应图

8.9.1　经典模拟滤波器的特征性质

8.9.1.1　巴特沃斯滤波器

低通巴特沃斯滤波器由下面的幅度平方频率响应来刻画：

$$|H(\omega)|^2 = \frac{1}{1+\left(\dfrac{\omega}{\omega_{\mathrm{p}}^{\mathrm{p}}}\right)^{2N}} \tag{8.24}$$

其中 N 是滤波器阶数，$\omega_{\mathrm{p}}^{\mathrm{p}}$ 是低通滤波器的 3 dB 截止频率（对于归一化的原型滤波器，$\omega_{\mathrm{p}}^{\mathrm{p}}=1$）。图8.12(a)描绘了典型的巴特沃斯低通滤波器的幅度-频率响应，可以看出它在通带和阻带都是单调的。这个响应是最平坦的，因为它初始时就是平坦的（在dc处的斜率为零）。

滤波器的阶数 N 由下式给定：

$$N \geqslant \frac{\log\left(\dfrac{10^{\frac{A_{\mathrm{s}}}{10}}-1}{10^{\frac{A_{\mathrm{p}}}{10}}-1}\right)}{2\log\left(\dfrac{\omega_{\mathrm{s}}^{\mathrm{p}}}{\omega_{\mathrm{p}}^{\mathrm{p}}}\right)} \tag{8.25}$$

其中 A_{p} 和 A_{s} 分别是用 dB 表示的通带波纹和阻带衰减，$\omega_{\mathrm{s}}^{\mathrm{p}}$ 是阻带的边沿频率。

归一化模拟巴特沃斯滤波器的传递函数 $H(s)$ 无限远处包含零点，极点均匀分布在 s 平面半径等于 1 的圆上，这些极点的位置如下所示（Stearns and Hush, 1990; Jong, 1982）：

$$s_k = \mathrm{e}^{\mathrm{j}\pi(2k+N-1)/2N} = \cos\left[\frac{(2k+N-1)\pi}{2N}\right] + \mathrm{j}\sin\left[\frac{(2k+N-1)\pi}{2N}\right],\quad k=1,2,\ldots,N \tag{8.26}$$

极点以复共轭对出现，位于 s 平面的左侧。

8.9.1.2　切比雪夫滤波器

切比雪夫特性提供了另外一种求合适的模拟传递函数 $H(s)$ 的方法。有类型I和类型II两种类型的切比雪夫滤波器，它们具有如下的特性（参见图8.12(b)和图8.12(c)）：

- 类型I，在通带有相等的波纹，在阻带单调变化；
- 类型II，在阻带有相等的波纹，在通带单调变化。

例如，类型I的切比雪夫滤波器是由下面的幅度平方响应来刻画的，

$$|H(\omega')|^2 = \frac{K}{1+\varepsilon^2 C_N^2(\omega'/\omega_{\mathrm{p}}')} \tag{8.27a}$$

其中 $C_N(\omega'/\omega_{\mathrm{p}}')$ 是切比雪夫多项式，它展示了通带内相等的波纹，N 除了表示滤波器的阶数之外，还表示多项式的阶数，ε 确定了通带波纹，它用 dB 表示如下：

$$\text{通带波纹} \leqslant 10\log_{10}(1+\varepsilon^2) = -20\log_{10}(1-\delta_{\mathrm{p}}) \tag{8.27b}$$

图8.12(b)给出了类型I的切比雪夫特性的典型幅度响应。对于切比雪夫响应来说，传递函数 $H(s)$ 依赖于期望的通带波纹和滤波器的阶数 N。滤波器的阶数 N 由下式给定：

$$N \geqslant \frac{\cosh^{-1}\left(\dfrac{10^{\frac{A_s}{10}}-1}{10^{\frac{A_p}{10}}-1}\right)}{\cosh^{-1}\left(\dfrac{\omega_s^p}{\omega_p^p}\right)} \tag{8.28}$$

其中A_p和A_s分别是用 dB 表示的通带波纹和阻带衰减，ω_s^p是阻带的边沿频率。

归一化切比雪夫 LPF 的极点位于s平面的椭圆上，坐标为（Stearns and Hush, 1990）

$$s_k = \sinh(\alpha)\cos(\beta_k) + \mathrm{j}\cosh(\alpha)\sin(\beta_k) \tag{8.29}$$

其中

$$\alpha = \frac{1}{N}\sinh^{-1}\left(\frac{1}{\varepsilon}\right);\ \beta_k = \frac{(2k+N-1)\pi}{2N},\quad k = 1, 2, \ldots, N$$

8.9.1.3　椭圆滤波器

椭圆滤波器在通带和阻带都表现出了等波纹特性，参见图 8.12(d)。它是由下面的幅度平方响应来刻画的，

$$|H(\omega')|^2 = \frac{K}{1+\varepsilon^2 G_N^2(\omega')} \tag{8.30}$$

其中$G_N(\omega')$是一个切比雪夫有理函数。和巴特沃斯、切比雪夫滤波器不一样，椭圆滤波器的极点没有简单的表达形式。极点位置的计算方法已经实现了（Antoniou, 1979; Jong, 1982; DeFatta, et al., 1988）。椭圆低通滤波器的零点是纯虚数。

椭圆特性根据幅度响应提供了最有效率的滤波器。对于一组给定的规范，椭圆特性产生的滤波器阶数最小。它应该是在 IIR 滤波器设计时首要选择的方法，除了要考虑相位响应时，此时巴特沃斯响应可能是首选的。

巴特沃斯、切比雪夫和椭圆特性的$H(s)$的多项式表在大多数的模拟设计著作中都有其归一化的形式，可以用在双线性变换中。不过实际上，由$H(s)$计算$H(z)$是通过一个软件包进行的，后面我们将介绍该软件包。

8.9.2　利用经典模拟滤波器的 BZT 方法

在原型低通滤波器不存在的情况下，BZT 法的步骤如下：

(1) 像前面描述的那样，对数字滤波器带沿频率或边界频率进行预弯曲。

(2) 根据数字滤波器规范和经典滤波器特性，找出一个合适的原型低通模拟滤波器。这包括利用某一频率变换方程（依赖于数字滤波器的类型 -LP、HP、BP 或者 BS），反过来确定原型 LP 滤波器的性能规范。由此求出原型滤波器的阶数和它的传递函数$H(s)$。

(3) 正如前面描述过的一样，对原型模拟 LP 滤波器的$H(s)$，通过频率转换和伸缩变换来反向归一化，从而得到一个新的传递函数$H'(s)$。

(4) 如前面描述过的一样，通过在频率伸缩后的传递函数$H(z)$中替换s，应用 BZT 得到期望的数字滤波器的传递函数$H'(s)$。

现在我们来看每一个滤波器类型的基本概念（LP、HP、BP 和 BS）。

8.9.2.1　低通滤波器——基本概念

从低通到低通的转换由下式给出（参见 8.21a 式）：

$$s = \frac{s}{\omega_p'}$$

如果我们在上式中用$j\omega$代替s，并且为了区别起见，用ω^p代表原型滤波器的频率，用ω_{lp}代表要设计的低通滤波器的频率，那么上式变为

$$j\omega^p = j\frac{\omega_{lp}}{\omega_p'}, \text{i.e. } \omega^p = \frac{\omega_{lp}}{\omega_p'} \tag{8.31}$$

8.31式定义了原型滤波器响应和我们希望设计的反向归一化低通滤波的频率之间的关系。给出了反向归一化低通滤波器的边界频率之后，我们利用8.31式来确定原型滤波器的边界频率以及它的性能规范。

原型滤波器的三个边界频率是：0，通带带沿频率；ω_p^p（实际上这通常是1）；阻带带沿频率ω_s^p。

(1) 当 $\omega_{lp}=0$ 时，$\omega^p=0$（从8.31式得出）

(2) 当 $\omega_{lp}=\omega_p'$ 时（即通带带沿频率），$\omega^p=\omega_p'/\omega_p'=1=\omega_p^p$

(3) 当 $\omega_{lp}=\omega_s'$时，$\omega^p=\omega_s'/\omega_p'=\omega_s^p$

因此，原型滤波器的边界频率是0、1、ω_s'/ω_p'。

图8.13给出了反向归一化低通滤波器和原型滤波器的频率之间的关系。

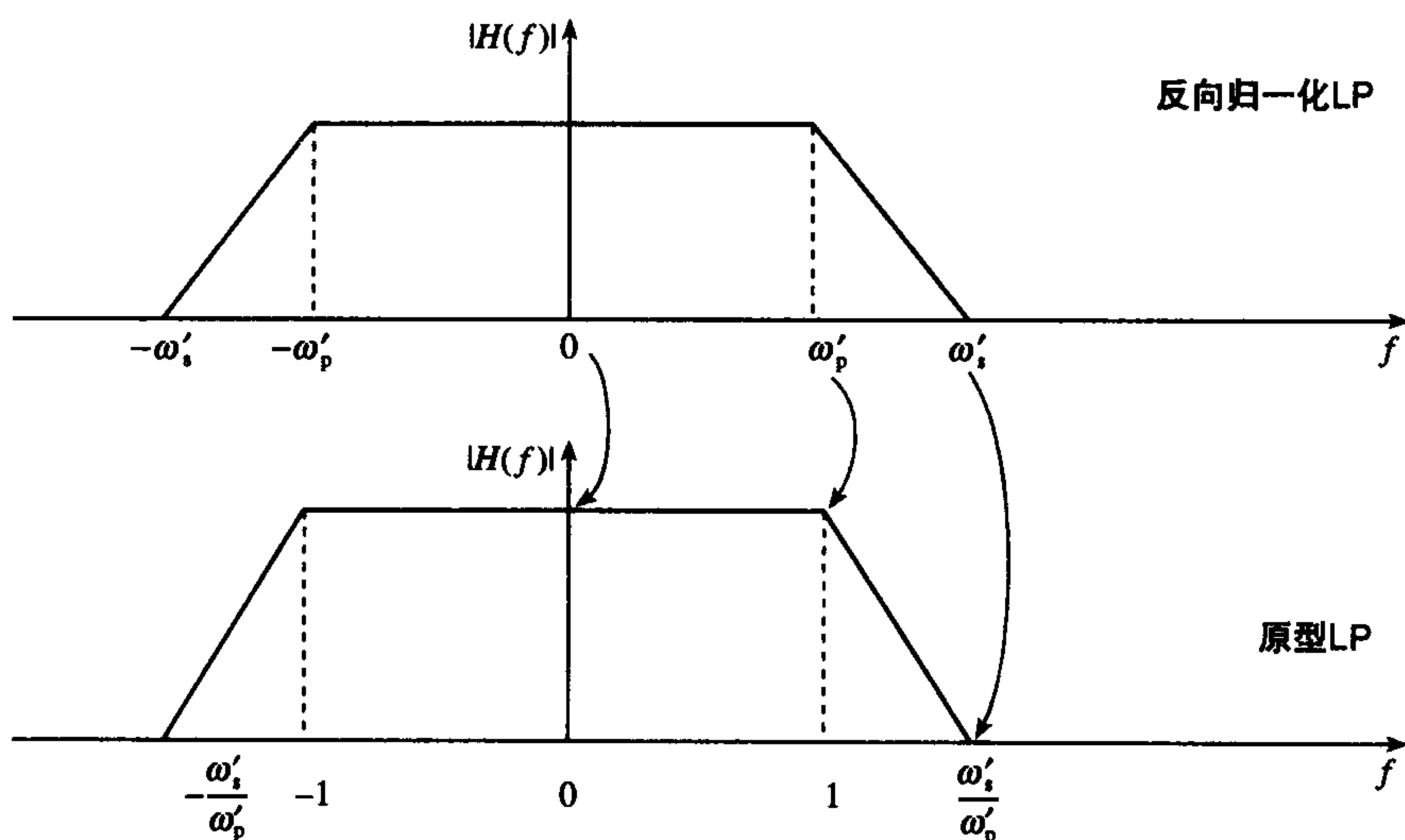


图8.13 反向归一化LP和原型LP滤波器中频率之间的关系

8.9.2.2 高通滤波器——基本概念

用ω_{hp}表示反向归一化高通滤波器的频率，用ω^p表示原型LP滤波器的频率（像以前一样），根据低通到高通的转换方程$s=\omega_p'/s$，原型LP滤波器和期望的高通滤波器的频率之间的关系可求得为

$$\omega^p = -\frac{\omega_p'}{\omega_{hp}} \tag{8.32}$$

根据期望的高通滤波器的边界频率，利用8.32式，我们现在可以指定原型LP滤波器的边界频率如下：

(1) 当 $\omega_{hp}=0$ 时，$\omega^{p}=\infty$（利用 8.32 式）

(2) 当 $\omega_{hp}=\omega'_{p}$ 时（即通带带沿频率），$\omega^{p}=-1$

(3) 当 $\omega_{hp}=\omega'_{s}$ 时，$\omega^{p}=-\dfrac{\omega'_{p}}{\omega'_{s}}$

(4) 当 $\omega_{hp}=-\omega'_{p}$ 时，$\omega^{p}=1$

(5) 当 $\omega_{hp}=-\omega'_{s}$ 时，$\omega^{p}=\dfrac{\omega'_{p}}{\omega'_{s}}$

因此，用来设计高通滤波器的原型低通滤波器的三个边界频率是 0、1 和 ω'_{p}/ω'_{s}。

图 8.14 绘出了原型 LP 滤波器的边界频率及其与反向归一化高通滤波器的频率之间的关系。在反向归一化的高通滤波器中，低通到高通的转换对频率的映射如下：它把零频率映射到无限远处，ω_{p} 映射到 1，无限远处映射到零。

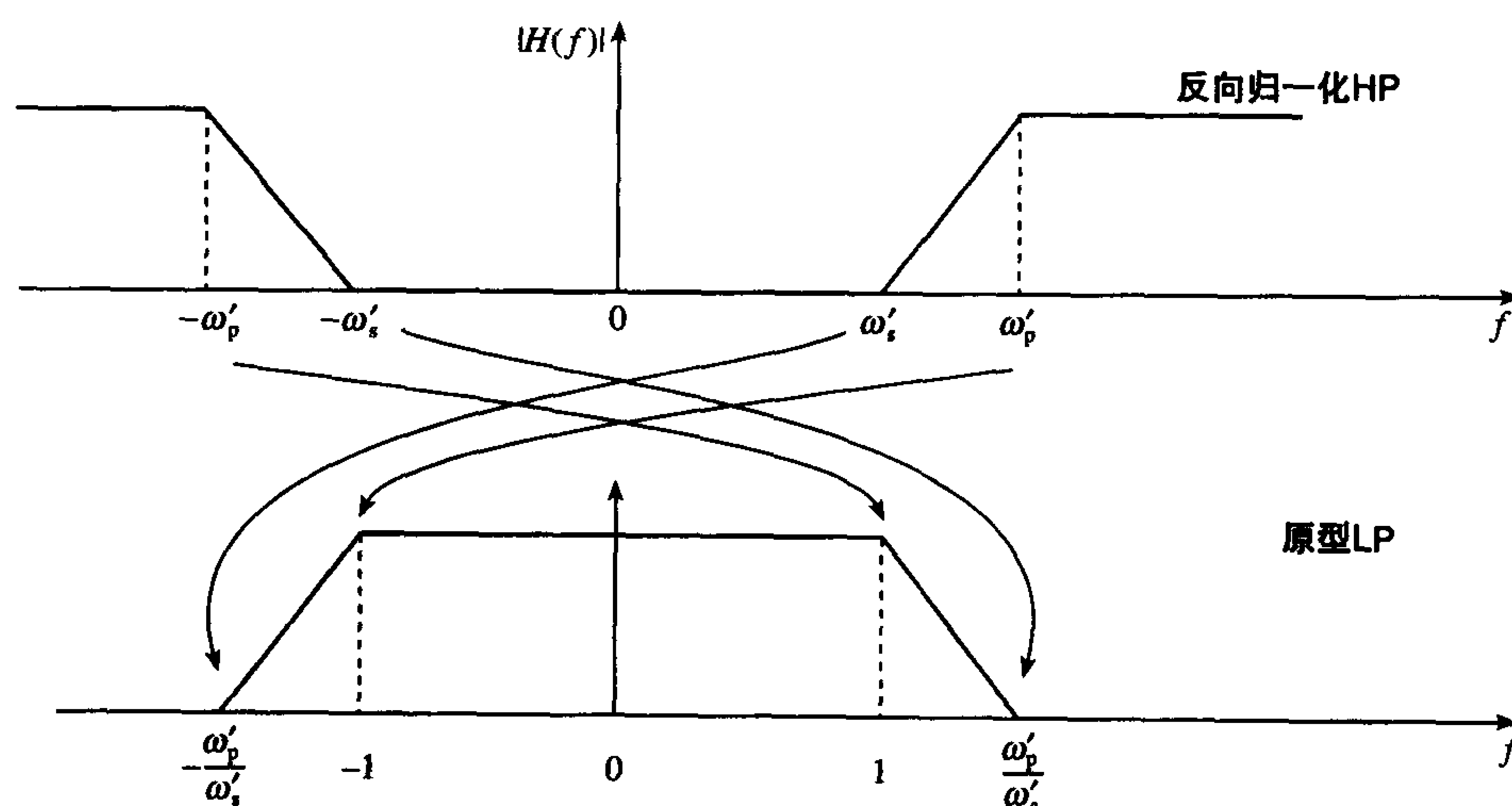


图 8.14 反向归一化 HP 滤波器和原型 LP 滤波器中频率之间的关系

8.9.2.3 带通滤波器——基本概念

从低通到带通的转换关系为

$$s=\frac{s^{2}+\omega_{0}^{2}}{Ws}$$

根据低通到带通的转换，带通滤波器的频率 ω_{bp} 和原型 LPF 的频率 ω^{p} 之间的关系为

$$j\omega^{p}=\frac{(j\omega_{bp})^{2}+\omega_{0}^{2}}{jW\omega_{bp}}$$

即

$$\omega^{p}=\frac{\omega_{bp}^{2}-\omega_{0}^{2}}{W\omega_{bp}} \tag{8.33}$$

带通滤波器有四个边界频率或带沿频率和一个中心频率：

ω'_{p1}, ω'_{p2} = 通道的下边沿和上边沿频率

$\omega'_{s1}, \omega'_{s2}$ = 阻带的下边沿和上边沿频率

ω_0 = 中心频率 ($\omega_0^2 = \omega'_{p1}\omega'_{p2}$)

利用8.33式给出的关系，原型LP滤波器的带沿频率可以根据带通滤波器的带沿频率求出：

(1) 当 $\omega_{bp} = \omega'_{s1}$ 时，$\omega^p = \omega'^p_{s1} = \dfrac{\omega'^2_{s1} - \omega_0^2}{W\omega'_{s1}}$

(2) 当 $\omega_{bp} = \omega'_{p1}$ 时，$\omega^p = \dfrac{\omega'^2_{p1} - \omega_0^2}{W\omega'_{p1}} = \dfrac{\omega'^2_{p1} - \omega'_{p1}\omega'_{p2}}{(\omega'_{p2} - \omega'_{p1})\omega'_{p1}} = -1$

(3) 当 $\omega_{bp} = \omega'_{p2}$ 时，$\omega^p = \dfrac{\omega'^2_{p2} - \omega_0^2}{W\omega'_{p2}} = \dfrac{\omega'^2_{p2} - \omega'_{p1}\omega'_{p2}}{(\omega'_{p2} - \omega'_{p1})\omega'_{p2}} = 1$

(4) 当 $\omega_{bp} = \omega'_{s2}$ 时，$\omega^p = \omega'^p_{s2} = \dfrac{\omega'^2_{s2} - \omega_0^2}{W\omega'_{s2}}$

(5) $\omega_{bp} = \omega_0$，$\omega^p = \dfrac{\omega'^2_0 - \omega_0^2}{W\omega_0^2} = 0$

(6) $\omega_s^p = \min(\omega'^p_{s1}, \omega'^p_{s2})$

因此，感兴趣的原型LP滤波器的边界频率是

$$0, 1, \min(\omega^p_{s1}, |\omega^p_{s2}|)$$

图8.15描绘了带通滤波器和原型LP滤波器之间的频率映射。我们注意到，例如，带通滤波器的中心频率映射到原型滤波器的零频率，通带上沿频率和阻带上沿频率 ω'_{p2} 和 ω'_{s2} 分别映射到原型滤波器的正的通带边沿频率和阻带边沿频率。另一方面，通带的下边沿频率和阻带的下边沿频率 ω'_{p1} 和 ω'_{s1}，分别映射到原型滤波器的负的通带边沿频率和负的阻带边沿频率。实践中，我们令原型滤波器的阻带边沿频率等于两个阻带频率 ω^p_{s1} 和 ω^p_{s2} 中较小的那个，这正如上面指出的一样。令原型LP滤波器通带波纹和阻带衰减等于数字带通滤波器通带波纹和阻带衰减。

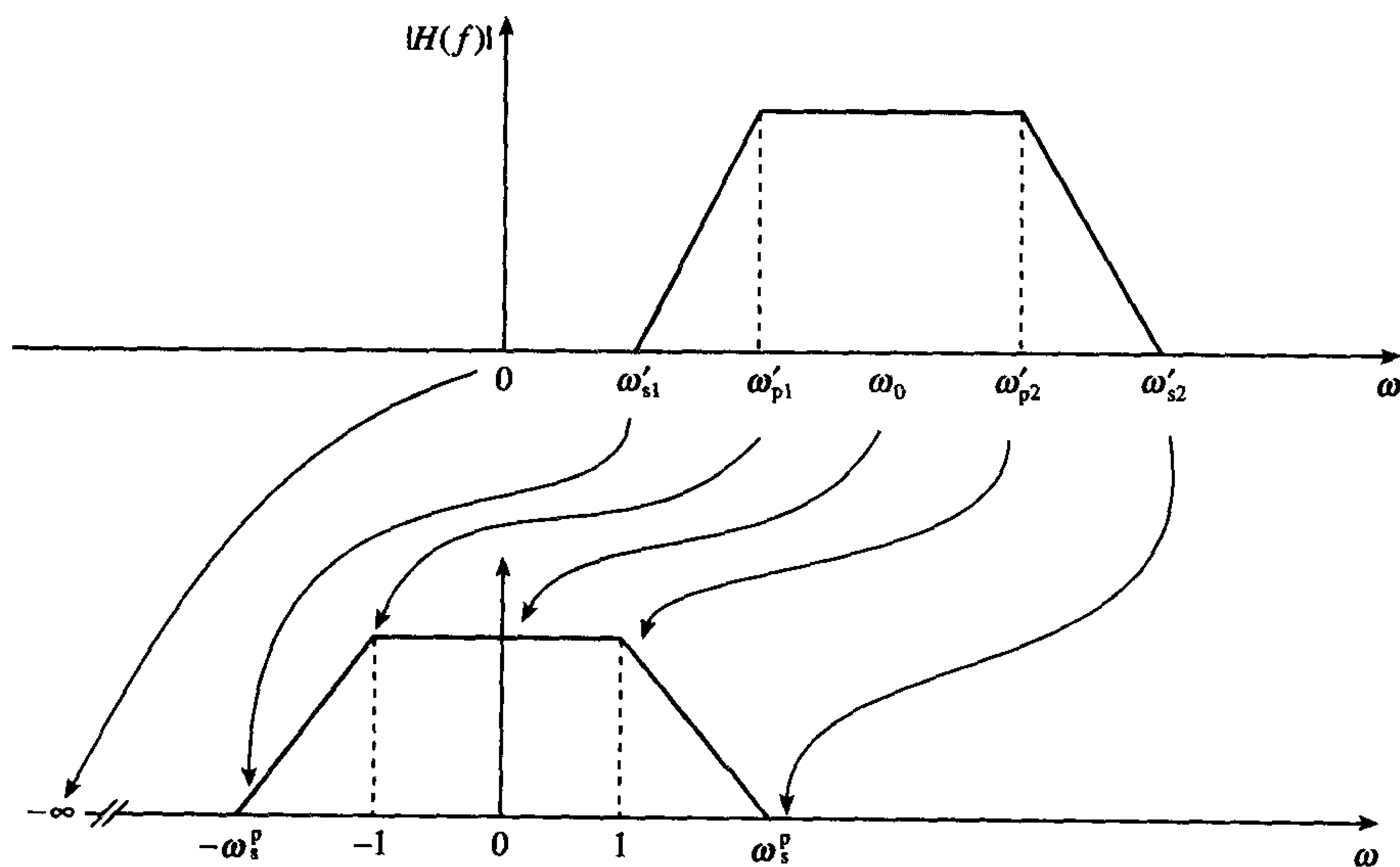


图8.15 原型LP到BPF的映射

根据原型LP滤波器的性能规范，我们利用8.25式或者8.28式可以确定滤波器的阶数N以及它的传递函数。带通滤波器的阶数是原型LP阶数的2倍，即$2N$。原型滤波器的极点可以从8.26式和8.29式求出。对于巴特沃斯和切比雪夫（类型I）滤波器，原型LP的零点在无穷远处，但是对于椭圆滤波器，零点为纯虚数。从极点和零点的位置（或者经典滤波器的标准表）可以得出原型滤波器的传递函数。

8.9.2.4　带阻滤波器——基本概念

从低通到带阻的转换由下式给出：

$$s = \frac{Ws}{s^2 + \omega_0^2}$$

带阻频率 ω_{bs} 和原型 LPF 的带阻频率 ω^p 有如下关系：

$$j\omega^p = \frac{jW\omega_{bs}}{(j\omega_{bs})^2 + \omega_0^2}$$

即

$$\omega^p = \frac{W\omega_{bs}}{\omega_0^2 - \omega_{bs}^2} \tag{8.34}$$

根据8.34式表示的关系，从期望的数字带阻滤波器的带沿频率，可以求出原型LP滤波器的带沿频率。回顾前面，带阻滤波器有四个带沿频率 $-\omega'_{p1}$、ω'_{p2}（下通带带沿频率和通带带沿频率）、ω'_{s1}、ω'_{s2}（下阻带带沿频率和上阻带带沿频率）和一个中心频率 ω_0（$\omega_0^2 = \omega'_{p1}\omega'_{p2}$）：

(1) 当 $\omega_{bs} = \omega'_{p1}$ 时，$\omega^p = \dfrac{W\omega'_{p1}}{\omega_0'^2 - \omega_{p1}'^2} = \dfrac{(\omega'_{p2} - \omega'_{p1})\omega'_{p1}}{\omega'_{p1}\omega'_{p2} - \omega_{p1}'^2} = 1$

(2) 当 $\omega_{bs} = \omega'_{s1}$ 时，$\omega^p = \omega_s^{p(1)} = \dfrac{W\omega'_{s1}}{\omega_0^2 - \omega_{s1}^2}$

(3) 当 $\omega_{bs} = \omega'_{s2}$ 时，$\omega^p = \omega_s^{p(2)} = \dfrac{W\omega'_{s2}}{\omega_0^2 - \omega_{s2}^2}$

(4) 当 $\omega_{bs} = \omega_0$ 时，$\omega^p = \dfrac{W\omega_0}{\omega_0^2 - \omega_0^2} = \infty$

(5) 当 $\omega_{bs} = \omega'_{p2}$ 时，$\omega^p = \dfrac{W\omega'_{p2}}{\omega_0^2 - \omega_{p2}'^2} = \dfrac{(\omega'_{p2} - \omega'_{p1})\omega'_{p2}}{\omega'_{p1}\omega'_{p2} - \omega_{p2}'^2} = -1$

因此，原型LP滤波器的阻带带沿频率 $\omega_s^p = \min(\omega_s^{p(1)}, \omega_s^{p(2)})$，通带带沿频率是1。通带波纹和阻带衰减分别是$A_p$和$A_s$。带阻滤波器的频率和原型低通滤波器的频率之间的映射如图8.16所示。我们看到带阻滤波器的带阻上沿频率和带通上边沿频率映射到原型滤波器的负频率上，而通带和阻带的下边沿频率映射到原型滤波器的正频率上。

感兴趣的原型LP滤波器的边界频率是

$$0,\ 1,\ \omega_s^p（其中\ \omega_s^p = \min(\omega_s^{p(1)}, \omega_s^{p(2)})）$$

从原型LP滤波器的性能规范出发，我们可以利用8.25式或8.28式求出滤波器的阶数N和传递函数。带通滤波器的阶数是原型LP的2倍，也就是$2N$。

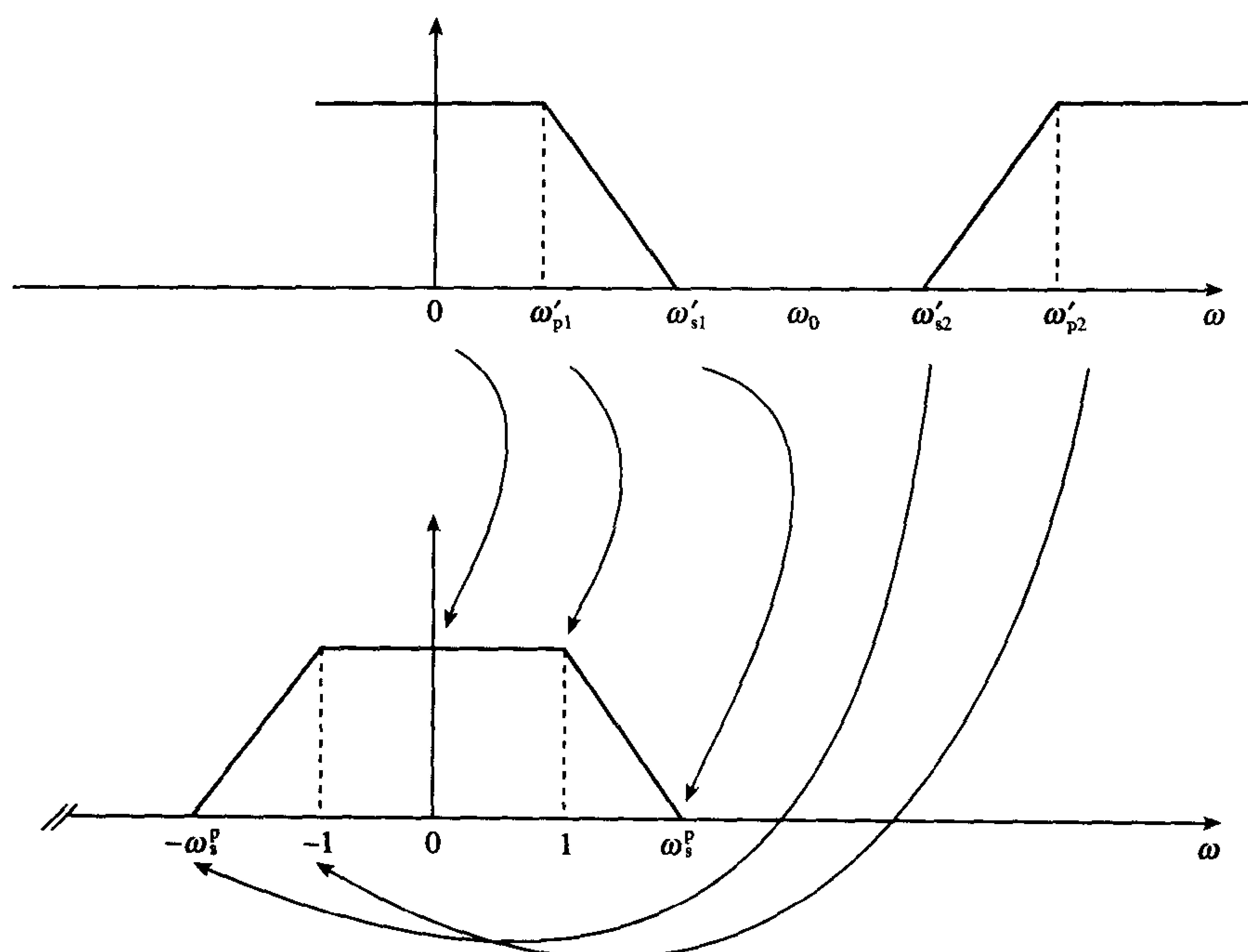


图 8.16 反向归一化 BS 和原型 LP 滤波器频率之间的关系

8.9.3 设计举例（低通、高通、带通和带阻滤波器）

例 8.11 低通滤波器 设计一个低通滤波器，要求满足如下性能规范：

通带	0 ~ 500 Hz
阻带	2 ~ 4 kHz
通带波纹	3 dB
阻带衰减	20 dB
抽样频率	8 kHz

(1) 求合适的模拟原型低通滤波器的通带和阻带的边沿频率；
(2) 求原型低通滤波器的阶数 N；
(3) 利用双线性 z 变换求离散时间滤波器的系数和传递函数。

假设滤波器具有巴特沃斯特性。

解：

(1) 从性能规范可知预扭曲的频率是

$$\omega'_{\rm p} = \tan\left(\frac{2\pi \times 500}{2 \times 8000}\right) = 0.198\,912$$

$$\omega'_{\rm s} = \tan\left(\frac{2\pi \times 2000}{2 \times 8000}\right) = 1$$

$$\omega_{\rm s}^{\rm p} = \frac{\omega'_{\rm s}}{\omega'_{\rm p}} = 1/0.198\,912 = 5.0273$$

因此，原型 LP 滤波器的预扭曲通带边沿和阻带边沿频率是 0、1、5.0273。

(2) 利用 8.25 式和以上的参数值来求滤波器的阶数。

现在

$$10^{A_s/10}-1=10^{20/10}-1=99;\ 10^{A_p/10}-1=10^{3/10}-1=0.995\,262\,3;$$

$$\log\left(\frac{99}{0.995\,262\,3}\right)=1.997\,697$$

对于原型 LPF，

$$\omega_p^p=1;\quad \omega_s^p=5.0273;\quad \log\left(\frac{\omega_s^p}{\omega_p^p}\right)=2\log(5.0273)=1.402\,66$$

$$N\geqslant\frac{1.997\,697}{1.402\,66}=1.424.\ \text{令}\ N=2$$

(3) 原型滤波器的极点是（由 8.26 式）

$$s_{p,1}=\cos\left[\frac{(2+2-1)\pi}{4}\right]+\mathrm{j}\sin\left[\frac{(2+2-1)\pi}{4}\right]=-\frac{\sqrt{2}}{2}+\mathrm{j}\frac{\sqrt{2}}{2}$$

$$s_{p,2}=-\frac{\sqrt{2}}{2}-j\frac{\sqrt{2}}{2}$$

s 平面的传递函数 $H(s)$ 是

$$H(s)=\frac{1}{(s-s_{p,1})(s-s_{p,2})}=\frac{1}{s^2+\sqrt{2}s+1}$$

频率伸缩后 s 平面的传递函数是

$$H'(s)=H(s)\Big|_{\frac{s}{\omega_p'}}=\frac{1}{\left(\dfrac{s}{\omega_p'}\right)^2+\sqrt{2}\dfrac{s}{\omega_p'}+1}$$

$$=\frac{\omega_p'^2}{s^2+\sqrt{2}s\omega_p'+\omega_p'^2}$$

应用 BZT：

$$H(z)=H'(s)\Big|_{s=\frac{z-1}{z+1}}=\frac{\omega_p'^2}{\left(\dfrac{z-1}{z+1}\right)^2+\sqrt{2}\omega_p'\left(\dfrac{z-1}{z+1}\right)+\omega_p'^2}$$

$$=\frac{\omega_p'^2(z+1)^2}{(z-1)^2+\sqrt{2}\omega_p'(z-1)(z+1)+\omega_p'^2(z+1)^2}$$

在化简和对上下都除以 z^2 后我们有

$$H(z)=\frac{\omega_p'^2}{1+\sqrt{2}\omega_p'+\omega_p'^2}\times\frac{1+2z^{-1}+z^{-2}}{1+\dfrac{2(\omega_p'^2-1)z^{-1}}{1+\sqrt{2}\omega_p'+\omega_p'^2}+\dfrac{(1-\sqrt{2}\omega_p'+\omega_p'^2)z^{-2}}{1+\sqrt{2}\omega_p'+\omega_p'^2}}$$

利用参数的值：

$$1+\sqrt{2}\omega_p'+\omega_p'^2=1.320\,87;\ \omega_p'^2-1=-0.960\,43$$

$$1-\sqrt{2}\omega_p'+\omega_p'^2=0.758\,285\,8;\ \omega_p'^2=0.039\,565\,9$$

代入上面的方程且化简，得

$$H(z)=\frac{0.029\,95(1+2z^{-1}+z^{-2})}{1-1.4542z^{-1}+0.574\,08z^{-2}}$$

例 8.12 高通滤波器 设计一个高通数字滤波器，要求的性能规范为

通带	2 ~ 4 kHz
阻带	0 ~ 500 Hz
通带波纹	3 dB
阻带衰减	20 dB
抽样频率	8 kHz

(1) 求合适的模拟原型低通滤波器的通带和阻带边沿频率；
(2) 求原型低通滤波器的阶数 N；
(3) 利用双线性 z 变换求离散时间滤波器的系数和传递函数。

假设滤波器具有巴特沃斯特性。

解：

(1) 由性能规范，预扭曲的频率是

$$\omega_s'=\tan\left(\frac{2\pi\times500}{2\times8000}\right)=0.198\,912$$

$$\omega_p'=\tan\left(\frac{2\pi\times2000}{2\times8000}\right)=1$$

$$\omega_s^p=\frac{\omega_p'}{\omega_s'}=1/0.198\,912=5.0273$$

因此，原型 LP 滤波器的通带和阻带边沿频率是 0、1、5.0273。

(2) 利用 8.25 式和上面的参数值来求滤波器的阶数。

现在

$$10^{A_s/10}-1=10^{20/10}-1=99;\ 10^{A_p/10}-1=10^{3/10}-1=0.995\,262\,3$$

$$\log\left(\frac{99}{0.995\,262\,3}\right)=1.997\,697$$

对原型 LPF，

$$\omega_p^p=1;\quad \omega_s^p=5.0273;\quad \log\left(\frac{\omega_s^p}{\omega_p^p}\right)=2\log(5.0273)=1.402\,66$$

$$N\geqslant\frac{1.997\,697}{1.402\,66}=1.424\ \ 令 N=2$$

原型滤波器的极点（由 8.26 式）为

$$s_{p,1}=\cos\left[\frac{(2+2-1)\pi}{4}\right]+\mathrm{j}\sin\left[\frac{(2+2-1)\pi}{4}\right]=-\frac{\sqrt{2}}{2}+\mathrm{j}\frac{\sqrt{2}}{2}$$

$$s_{p,2}=-\frac{\sqrt{2}}{2}-\mathrm{j}\frac{\sqrt{2}}{2}$$

s 平面传递函数 $H(s)$ 为

$$H(s)=\frac{1}{(s-s_{p,1})(s-s_{p,2})}=\frac{1}{s^2+\sqrt{2}s+1}$$

频率伸缩后，s 传递函数为

$$H'(s)=H(s)\Big|_{\frac{\omega_p'}{s}}=\frac{1}{\left(\frac{\omega_p'}{s}\right)^2+\sqrt{2}\frac{\omega_p'}{s}+1}$$

$$=\frac{s^2}{s^2+\sqrt{2}s\omega_p'+\omega_p'^2}$$

应用 BZT：

$$H(z)=H'(s)\Big|_{s=\frac{z-1}{z+1}}=\frac{\left(\frac{z-1}{z+1}\right)^2}{\left(\frac{z-1}{z+1}\right)^2+\sqrt{2}\omega_p'\left(\frac{z-1}{z+1}\right)+\omega_p'^2}$$

$$=\frac{(z-1)^2}{(z-1)^2+\sqrt{2}\omega_p'(z-1)(z+1)+\omega_p'^2(z+1)^2}$$

在化简和上、下都除以 z^2 后，我们有

$$H(z)=\frac{1}{1+\sqrt{2}\omega_p'+\omega_p'^2}\times\frac{1-2z^{-1}+z^{-2}}{1+\frac{2(\omega_p'^2-1)z^{-1}}{1+\sqrt{2}\omega_p'+\omega_p'^2}+\frac{(1-\sqrt{2}\omega_p'+\omega_p'^2)z^{-2}}{1+\sqrt{2}\omega_p'+\omega_p'^2}}$$

利用参数值：

$$1+\sqrt{2}\omega_p'+\omega_p'^2=3.414\,21;\ \omega_p'^2-1=0$$

$$1-\sqrt{2}\omega_p'+\omega_p'^2=0.585\,786;\ \omega_p'^2=1$$

代入上面的等式且化简，我们有

$$H(z)=\frac{0.292\,89(1-2z^{-1}+z^{-2})}{1+0.171\,57z^{-2}}$$

例 8.13　带通滤波器　设计一个具有巴特沃斯幅度－频率响应的带通数字滤波器，要求满足如下的性能规范：

通带的下边沿频率	200 Hz
通带的上边沿频率	300 Hz
阻带的下边沿频率	50 Hz
阻带的上边沿频率	450 Hz
通带波纹	3 dB
阻带衰减	20 dB
抽样频率	1 kHz

(1) 求合适的原型低通滤波器的通带和阻带的边沿频率；

(2) 求原型低通滤波器的阶数 N；

(3) 利用 BZT 法求离散时间滤波器的系数和传递函数。

解：

带通滤波器的预弯曲边界频率是

$$\omega_0 = 1; \qquad W = 0.6498$$

$$\omega'_{p1} = 0.7265; \qquad \omega'_{p2} = 1.376\,38$$

$$\omega'_{s1} = 0.1584; \qquad \omega'_{s2} = 6.3138$$

因此，原型LP滤波器的带沿频率是（利用上面的关系）

$$\omega_p^p = 1; \qquad \omega_s^p = 9.4721$$

因此，我们要求的原型LPF具有$\omega_p^p = 1; \omega_s^p = 9.4721, A_p = 3\text{ dB}; A_s = 20\text{ dB}$。

由8.25式，原型LPF的阶数可求得为

$$10^{A_s/10} - 1 = 10^{20/10} - 1 = 99;\ 10^{A_p/10} - 1 = 10^{3/10} - 1 = 0.995\,262\,3;$$

$$\log\left(\frac{99}{0.995\,262\,3}\right) = 1.997\,697\,6;\ \frac{\omega_s^p}{\omega_p^p} = 9.4721;\ \ 2\log(9.4721) = 1.952\,89$$

$$N = \frac{1.997\,697\,6}{1.952\,89} = 1.0229$$

N必须是一个整数，为简单起见我们令$N = 1$。对于一阶原型LP滤波器，s平面的传递函数为

$$H(s) = \frac{1}{s+1}$$

利用表上从低通到带通的传递函数，我们得到

$$H'(s) = H(s)\Big|_{s=\frac{s^2+\omega_0^2}{Ws}} = \frac{1}{\left(\dfrac{s^2+\omega_0^2}{Ws}\right)+1}$$

$$= \frac{Ws}{s^2 + Ws + \omega_0^2}$$

应用BZT，得

$$H(z) = H'(s)\Big|_{s=\frac{z-1}{z+1}}$$

$$= \frac{W\left(\dfrac{z-1}{z+1}\right)}{\left(\dfrac{z-1}{z+1}\right)^2 + W\left(\dfrac{z-1}{z+1}\right) + \omega_0^2}$$

代入ω_0^2和W的值，简化后有

$$H(z) = \frac{0.2452(1 - z^{-2})}{1 + 0.5095z^{-2}}$$

例8.14　带阻滤波器　设计一个具有巴特沃斯幅度－频率响应的带阻IIR滤波器，要求它满足如下的性能描述：

通带下沿	0 ~ 50 Hz
通带上沿	450 ~ 500 Hz

阻带	200 ~ 300 Hz
通带波纹	3 dB
阻带衰减	20 dB
抽样频率	1 kHz

(1) 求合适的原型低通滤波器的通带和阻带的边沿频率；

(2) 求原型低通滤波器的阶数 N；

(3) 利用 BZT 法求离散时间滤波器的系数和传递函数。

解：

带通滤波器的预扭曲的边界频率是

$$\omega_0 = 1; \qquad W = 6.1554$$
$$\omega'_{p1} = 0.1584; \qquad \omega'_{p2} = 6.3138$$
$$\omega'_{s1} = 0.7265; \qquad \omega'_{s2} = 1.376\,38$$

因此，原型滤波器的带沿频率是

$$\omega_p^p = 1; \qquad \omega_s^p = 9.4721$$

我们要求一个具有$\omega_p^p = 1$、$\omega_s^p = 9.4721$、$A_p = 3$ dB 和$A_s = 20$ dB 的原型 LPF。

从 8.25 式得到的原型 LPF 的阶数为

$$10^{A_s/10} - 1 = 10^{20/10} - 1 = 99;\ 10^{A_p/10} - 1 = 10^{3/10} - 1 = 0.995\,262\,3$$

$$\log\left(\frac{99}{0.995\,262\,3}\right) = 1.997\,697\,6; \quad \frac{\omega_s^p}{\omega_p^p} = 9.4721; \quad 2\log(9.4721) = 1.952\,89$$

$$N = \frac{1.997\,697\,6}{1.952\,89} = 1.0229$$

N必须是一个整数，为简化起见，我们令$N = 1$。一阶 LP 滤波器的s平面的传递函数如下给定：

$$H(s) = \frac{1}{s+1}$$

利用表上的从低通到带阻的转换函数我们得到

$$H'(s) = H(s)\Big|_{s=\frac{Ws}{s^2+\omega_0^2}} = \frac{1}{\left(\dfrac{Ws}{s^2+\omega_0^2}\right)+1}$$

$$= \frac{s^2+\omega_0^2}{s^2+Ws+\omega_0^2}$$

应用 BZT，得

$$H(z) = H'(s)\Big|_{s=\frac{z-1}{z+1}}$$

$$= \frac{\left(\dfrac{z-1}{z+1}\right)^2 + \omega_0^2}{\left(\dfrac{z-1}{z+1}\right)^2 + W\left(\dfrac{z-1}{z+1}\right) + \omega_0^2}$$

代入ω_0^2和W的值并简化，我们有

$$H(z)=\frac{0.2452(1+z^{-2})}{1-0.5095z^{-2}}$$

例8.15 求一个低通数字滤波器的传递函数，要求满足下面的性能规范：

通带	0～60 Hz
阻带	> 85 Hz
阻带衰减	> 15 dB

假设抽样频率是256 Hz，且具有巴特沃斯特性。

解：

这个例子解释了如何像8.23式所建议的那样，在BZT过程中把步骤4和5合并成一个。

(1) 数字滤波器的边界频率是

$$\omega_1 T=\frac{2\pi f_1}{F_s}=\frac{2\pi 60}{256}=2\pi\times 0.2344$$

$$\omega_2 T=\frac{2\pi f_2}{F_s}=\frac{2\pi 85}{256}=2\pi\times 0.3320$$

(2) 预弯曲的等价的模拟频率是

$$\omega_1'=\tan\left(\frac{\omega_1 T}{2}\right)=0.906\,347;\ \omega_2'=\tan\left(\frac{\omega_2 T}{2}\right)=1.715\,80$$

(3) 接下来我们需要求$H(s)$，它具有巴特沃斯特性，3 dB的截止频率是0.906 347，在85 Hz处的响应被降低15 dB。从8.25式出发，对于15 dB的衰减和3 dB的通带波纹，$N=2.68$。因为N必须是个整数，我们令$N=3$。给出的归一化的三阶滤波器为

$$H(s)=\frac{1}{(s+1)(s^2+s+1)}=\frac{1}{s+1}\,\frac{1}{s^2+s+1}$$
$$=H_1(s)\,H_2(s)$$

$$\cot\left(\frac{\omega_1 T}{2}\right)=\cot\left(\frac{2\pi\times 0.2344}{2}\right)=1.103\,155$$

在两个步骤里进行这个转换，一是对上面$H(s)$里的每一个因子，我们得到

$$H_2(z)=H_2(s)\big|_{s=\cot(\omega_1 T/2)[(z-1)/(z+1)]}$$
$$=0.3012\,\frac{1+2z^{-1}+z^{-2}}{1-0.1307z^{-1}+0.3355z^{-2}}$$

这是我们经过相当多的处理后得到的。类似地，我们得到$H_1(z)$为

$$H_1(z)=0.4754\,\frac{1+z^{-1}}{1-0.0490z^{-1}}$$

那么$H_1(z)$和$H_2(z)$可以合并，给出期望的传递函数$H(z)$：

$$H(z)=H_1(z)H_2(z)=0.1432\,\frac{1+3z^{-1}+3z^{-2}+z^{-3}}{1-0.1801z^{-1}+0.3419z^{-2}-0.0165z^{-3}}$$

8.10 通过映射 s 平面极点和零点来计算 IIR 滤波器的系数

8.10.1 基本概念

计算实际的IIR滤波器$H(z)$系数，一个可选的、或许更强大和灵活的方法是：把合适的模拟滤波器的每一个极点和零点，从s平面映射到z平面，接着由z平面的极点和零点来推导数字滤波器的系数。这个方法已被许多商业软件所采用，当滤波器的阶数比较高时，这个方法很具有吸引力。

通过映射s平面的极点、零点到z平面来计算 IIR 系数的过程总结如下。

8.10.1.1 步骤 1

如前面一样，设计者从归一化的N阶模拟低通滤波器出发，滤波器是巴特沃斯、切比雪夫或者椭圆类型，依设计要求而定。如果是巴特沃斯型则利用8.26式，如果是切比雪夫型则利用8.29式，得到归一化LPF的极点。椭圆类型的每一个极点都是复数，一般来说具有如下形式：

$$s_{1,k} = \alpha_{p,k} + j\beta_{p,k} \tag{8.35}$$

对于巴特沃斯和切比雪夫（类型I）的滤波器，原型LPF的零点在无穷远处；但对椭圆型滤波器，零点则是纯虚数的。一般来说，归一化LPF的零点的位置比极点更容易确定。

8.10.1.2 步骤 2

其次，归一化模拟LPF利用8.21a式~8.21d式中某个合适的转换式子，转化成LP、HP、BP或者BSF。

低通和高通滤波器

对于低通或者高通的数字滤波器，归一化LP的N个极点如下式那样转换（参见8.21a式和8.21b式）：

$$s_{1,k} = s_{1,k}/\omega'_p \quad k = 1, 2, \ldots, N \quad \text{低通到低通} \tag{8.36a}$$

$$s_{h,k} = \omega'_p/s_{1,k} \quad k = 1, 2, \ldots, N \quad \text{低通到高通} \tag{8.36b}$$

其中ω'_p是期望的通带边沿频率，$s_{1,k}$是模拟低通滤波器的极点，$s_{h,k}$是模拟高通滤波器的极点。

8.36a式和8.36b式之间的相似性是很显然的，这是因为低通和高通特性之间的对偶性引起的。当N为偶数时，将有$N/2$个复数极点对。当N为奇数时，将有$(N-1)/2$个复极点对和一个实极点。

对于经典滤波器——巴特沃斯、切比雪夫和椭圆，8.36a式和8.36b式的转换把原型LPF的零点映射到s平面的虚轴。在巴特沃斯或者切比雪夫滤波器里，原型滤波器的零点在无穷远处。在两种情况下，转换都是将零点从无穷远处映射到无穷远处（对于低通滤波器），或者从无穷远处映射到原点（对于高通滤波器），就像图8.17(a)(ii)和图8.17(b)(ii)表现的那样。

带通滤波器和带阻滤波器

对于带通数字滤波器，模拟BPF的极点是利用转换式从归一化原型LPF的极点得到的，

$$s_{1,k} = \frac{s_{b,k}^2 + \omega_0^2}{W s_{b,k}} \tag{8.37}$$

其中$s_{1,k}$是原型模拟LPF的极点，$s_{b,k}$是中间的模拟BPF的极点对，$W = \omega'_2 - \omega'_1$是滤波器通带的带宽，$\omega_0^2 = \omega'_1\omega'_2$给出了通带的中心频率。8.37式产生了如下的$s_{b,k}$的二次方程：

$$s_{b,k}^2 - W s_{1,k} s_{b,k} + \omega_0^2 = 0 \tag{8.38}$$

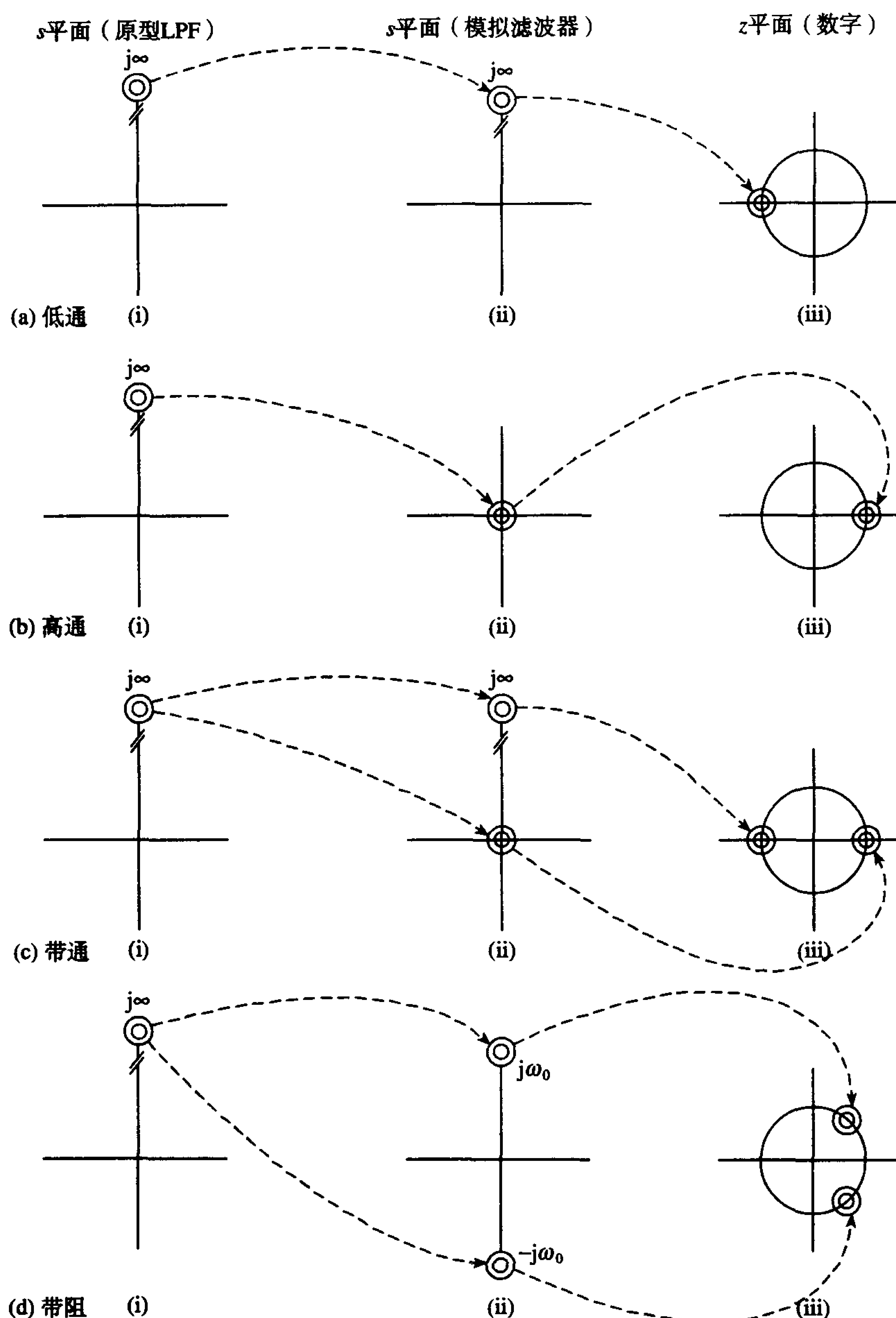


图 8.17 由二阶原型低通滤波器的零点映射到(a)、(b)、(c)、(d)

解出 $s_{b,k}$，推导出带通模拟滤波器极点的如下表达式：

$$s_{b,k}=\frac{W}{2}\left[s_{1,k}\pm\left(s_{1,k}^2-\frac{4\omega_0^2}{W^2}\right)^{1/2}\right] \tag{8.39}$$

从8.39可以看出，因为在变换式中有s^2项，所以每一个模拟LP极点$s_{1,k}$得出一对模拟BPF极点。一般来说，模拟低通滤波器的极点$s_{1,k}$是复的，那么平方根项也是复的。利用实数运算来计算复数的平方根，如果想得到正确的结果需要注意一些问题（参见附录 8C）。

对于带阻数字滤波器，下面的从低通到带阻的转换可以采用：

$$s_{1,k} = \frac{Ws_{r,k}}{s_{r,k}^2 + \omega_0^2} \tag{8.40}$$

其中 $s_{1,k}$ 是模拟原型 LPF 的极点，$s_{r,k}$ 是中间的模拟带阻滤波器的极点，W 是阻带的带宽，ω_0^2 是阻带的中心频率。为了从原型 LPF 得到模拟带阻滤波器的极点，8.40 式导出了下面的表达式：

$$s_{r,k} = \frac{W}{2}\left[s_{1,k}^{-1} \pm \left(s_{1,k}^{-2} - \frac{4\omega_0^2}{W^2}\right)^{1/2}\right] \tag{8.41}$$

不必吃惊，8.41 式除了极点是 LPF 的极点的倒数之外，和 8.39 式具有相同的形式，这是因为 BPF 和 BSF 的对偶性。

在低通和高通滤波器情形下，8.39 式和 8.41 式的转换是将零点映射到虚轴上。对于巴特沃斯和切比雪夫滤波器，从低通到带通的转换，是将原型低通滤波器的 N 个零点，从无穷远处映射到 s 平面的无穷远处和原点处（参见图 8.17(c)）。另一方面，从低通到带阻的转换，是将原型低通滤波器的 N 个零点，从无穷远处映射到 s 平面的 $\pm j\omega_0$；参见图 8.17(d)。对于椭圆带通和带阻滤波器，转换是将原型 LPF 的零点（它是纯虚数）映射到 $j\omega$ 轴的其他点。

8.10.1.3　步骤 3

接下来利用 BZT，把 s 平面的极点和零点映射到数字的 z 平面。每一个 s 平面的极点 $s_{p,k}$ 的映射如下式所示：

$$z_{p,k} = \frac{1 + s_{p,k}}{1 - s_{p,k}} \tag{8.42}$$

类似的，转换后的模拟滤波器的每一个 s 平面的零点 $s_{z,k}$，如下式映射到 z 平面：

$$z_{z,k} = \frac{1 + s_{z,k}}{1 - s_{z,k}} \tag{8.43}$$

图 8.17(a) ~ 图 8.17(d)解释了对巴特沃斯和切比雪夫滤波器、通过 BZT 把零点从 s 平面映射到 z 平面的方法。注意，例如，对于低通滤波器，零点（如图 8.17(a)(ii)所示）在 s 平面的无穷远处，8.43 式的 BZT 把这些零点映射到 z 平面的点 $z = -1$（如图 8.17(a)(iii)所示）；而对于带通滤波器，s 平面的零点（如图 8.17(c) (ii)所示）从原点映射到 $z = 1$，从无穷远处映射到 $z = -1$（如图 8.17(a) (iii)所示）。

对于椭圆低通、高通、带通和带阻滤波器，$s_{z,k}$ 是虚数，BZT 把这些零点映射到 z 平面的单位圆上。一般来说，通过 BZT 得到的所有经典滤波器（巴特沃斯，切比雪夫和椭圆）的 z 平面的零点都位于单位圆上，而无论什么样的滤波器类型。最后得到的经典滤波器的 $H(z)$ 的分子的系数总是整数（$0, \pm 1, \pm 2$）。

8.10.1.4　步骤 4

最后一步是确定二阶或一阶滤波器部分的分子和分母的系数。这是通过将复共轭极点和零点按下式进行合并得到的：

$$H_i(z) = \frac{(z - z_{z,k})(z - z_{z,k}^*)}{(z - z_{p,k})(z - z_{p,k}^*)} \tag{8.44a}$$

$$= \frac{1 + b_{1i}z^{-1} + b_{2i}z^{-2}}{1 + a_{1i}z^{-1} + a_{2i}z^{-2}} \tag{8.44b}$$

一般来说，位于 $\alpha \pm j\beta$ 的复极点或零点对可推导出 z 的二次方形式：

$$[z-(\alpha+\mathrm{j}\beta)][z-(\alpha-\mathrm{j}\beta)] = z^2 - 2\alpha z + \alpha^2 + \beta^2 = 1 - 2\alpha z^{-1} + (\alpha^2+\beta^2)z^{-2} \quad (8.45a)$$

实轴上的单极点或零点（也即在 $z=\pm\alpha$ 处）可推导出一个一阶因式形式：

$$z \pm \alpha = 1 \pm \alpha z^{-1} \quad (8.45b)$$

我们经常发现一个 N 阶滤波器在 z 平面上有 N 个实零点在实轴上。在这种情况下，我们可以把 z 平面的零点组成对，以至于每一个滤波器部分的分子是一个二次方形式：

$$1 \pm 2z^{-1} + z^{-2} \quad (8.45c)$$

整个传递函数 $H(z)$ 为

$$H(z) = KH_1(z)H_2(z)\ldots H_M(z)$$

其中 K 是一个增益因子，它用来调整通带的幅度响应达到期望的电平。在大多数情况下，K 被设为一个值，使得通带的最大响应等于1。

8.10.2 设计举例

例8.16 在某个DSP应用里，要求一个二阶数字带通滤波器（BPF），它具有巴特沃斯特性，通带是200～300 Hz，抽样频率是2 kHz。通过把一个合适的原型模拟低通滤波器的 s 平面的极点和/或零点映射到 z 平面来确定数字滤波器的传递函数。

画出并标出原型LPF、中间过渡的模拟BPF和数字BPF的极零图。

解：

要求一个一阶归一化LPF，因为带通转换使滤波器的阶数翻倍。因此，

$$H(s) = \frac{1}{s+1}$$

这个传递函数有一个单极点在 $s_{1,1}=-1$。现在，$F_s = 2\text{ kHz} = 1/T$。因此，预扭曲的带沿频率是

$$\omega_1' = \tan\left(\frac{\omega_1 T}{2}\right) = \tan\left(\frac{2\pi\times 200}{2\times 2000}\right) = 0.3249$$

$$\omega_2' = \tan\left(\frac{\omega_2 T}{2}\right) = \tan\left(\frac{2\pi\times 300}{2\times 2000}\right) = 0.5095$$

因此 ω_0 和 W 为

$$\omega_0^2 = \omega_1'\omega_2' = 0.1655,\ W = \omega_2' - \omega_1' = 0.1846$$

LPF的单极点利用8.39式转换成BPF的两个极点，如下式所示：

$$s_{b,1} = \frac{0.1846}{2}\left\{-1 + \left[(-1)^2 - \frac{4\times 0.1655}{(0.1846)^2}\right]^{1/2}\right\}$$

$$= -0.0923 + 0.4172\mathrm{j}$$

$$s_{b,2} = \frac{0.1846}{2}\left\{-1 - \left[(-1)^2 - \frac{4\times 0.1655}{(0.1846)^2}\right]^{1/2}\right\}$$

$$= -0.0923 - 0.4172\mathrm{j} = s_{b,1}^*$$

根据 8.42 式的 BZT 转换，我们有

$$z_{p,1} = \frac{1 - 0.0923 + 0.4172j}{1 + 0.0923 - 0.4172j} = 0.5979 + 0.6103j$$

$$z_{p,2} = z^*_{p,1}$$

原型LPF在无穷远处有一个单零点。通过低通到带通的转换，它被映射到s平面的原点和无穷远处。也就是$s_{z,1} = 0, s_{z,2} = \infty$。BZT 把这些零点映射到$z$平面的$z = 1$和$z = -1$。

$$s_{z,1} \to z_{z,1} = 1; \quad s_{z,2} \to z_{z,2} = -1$$

我们现在可以由极点和零点来确定离散传递函数$H(z)$，

$$\begin{aligned} H(z) &= \frac{(z-1)(z+1)}{(z - z_{p,1})(z - z_{p,2})} \\ &= \frac{z^2 - 1}{z^2 - 1.1958z + 0.7995} = \frac{1 - z^{-2}}{1 - 1.1958z^{-1} + 0.7995z^{-2}} \end{aligned}$$

传递函数除了差一个常数因子之外，和例 8.9 里的传递函数相同。极零图也与图 8.11 给出的相同。

例 8.17　从合适的模拟LPF出发，求因式分解形式的切比雪夫数字HPF的传递函数，它满足下面的性能规范：

通带边沿频率	15 kHz
在 18 kHz 的衰减	> 30 dB
通带波纹	1 dB
抽样频率	48 kHz

解：

预扭曲的边界频率是

$$\omega'_p = \tan\left(\frac{15\,000\pi}{48\,000}\right) = 1.4966$$

$$\omega'_s = \tan\left(\frac{18\,000\pi}{48\,000}\right) = 2.4142$$

从通带的性能规范有$\varepsilon = 0.3493$。适当的 LP 切比雪夫滤波器的阶数是 5（最接近的整数）。

利用$\alpha = 1/N \sinh^{-1}(1/\varepsilon) = 0.3548$，$\sinh(\alpha) = 0.3623$和$\cosh(\alpha) = 1.0636$，归一化低通切比雪夫滤波器的左边的极点位于（$\omega'_p = 1$）：

$$\begin{aligned} s_{1,1} &= 0.3623\cos\left[\frac{(2+5-1)\pi}{10}\right] + 1.0636\sin\left[\frac{(2+5-1)\pi}{10}\right]j \\ &= -0.111\,96 + 1.0115j \end{aligned}$$

$$\begin{aligned} s_{1,2} &= 0.3623\cos\left[\frac{(4+5-1)\pi}{10}\right] + 1.0636\sin\left[\frac{(4+5-1)\pi}{10}\right]j \\ &= -0.2931 + 0.6252j \end{aligned}$$

$$s_{1,3} = 0.3623\cos\left[\frac{(6+5-1)\pi}{10}\right] + 1.0636\sin\left[\frac{(6+5-1)\pi}{10}\right]\mathrm{j} = -0.3623$$

$$s_{1,4} = 0.3623\cos\left[\frac{(8+5-1)\pi}{10}\right] + 1.0636\sin\left[\frac{(8+5-1)\pi}{10}\right]\mathrm{j}$$

$$= -0.2931 - 0.6252\mathrm{j}$$

$$s_{1,5} = 0.3623\cos\left[\frac{(10+5-1)\pi}{10}\right] + 1.0636\sin\left[\frac{(10+5-1)\pi}{10}\right]\mathrm{j}$$

$$= -0.111\,96 - 1.0115\mathrm{j}$$

读者应该注意到极点分布的对称性，$s_{1,1}$ 和 $s_{1,5}$ 形成了一个复共轭对，$s_{1,2}$ 和 $s_{1,4}$ 也是如此。另外我们也注意到，每一个极点都位于 s 平面的左边，这是滤波器稳定的必要条件。

利用 8.36b 式将原型低通滤波器的极点转换到期望的 HP 的极点：

$$s_{h,1} = -0.1618 - 1.4616\mathrm{j}$$

$$s_{h,2} = -0.920\,13 - 1.9625\mathrm{j}$$

$$s_{h,3} = -4.1306$$

$$s_{h,4} = s^*_{h,2} = -0.920\,13 + 1.9625\mathrm{j}$$

$$s_{h,5} = s^*_{h,1} = -0.1618 + 1.4616\mathrm{j}$$

接下来，利用 BZT 将极点从 s 平面映射到 z 平面。从 8.42 式有，经过 BZT 后的 z 平面极点如下式给定（仅考虑那些在实轴上的极点）：

$$z_{h,1} = -0.3335 + 0.8386\mathrm{j}$$

$$z_{h,2} = -0.4906 + 0.5207\mathrm{j}$$

$$z_{h,3} = -0.6102$$

所有的零点 $z_{z,k}$ 都位于 $z=1$。那么二阶或一阶滤波器部分的系数可以从极点和零点得到（8.45a 式和 8.45b 式），

$$b_{11} = -2 \qquad b_{12} = -2 \qquad b_{13} = -1$$

$$b_{21} = 1 \qquad b_{22} = 1 \qquad b_{23} = 0$$

$$a_{11} = 0.6670 \qquad a_{12} = 0.9812 \qquad a_{13} = 0.6102$$

$$a_{21} = 0.8145 \qquad a_{22} = 0.5118 \qquad a_{23} = 0$$

最后，可得到传递函数：

$$H(z) = KH_1(z)H_2(z)H_3(z)$$

其中

$$H_1(z) = \frac{1 - 2z^{-1} + z^{-2}}{1 + 0.6670z^{-1} + 0.8145z^{-2}}$$

$$H_2(z) = \frac{1 - 2z^{-1} + z^{-2}}{1 + 0.9812z^{-1} + 0.5118z^{-2}}$$

$$H_3(z) = \frac{1 - z^{-1}}{1 + 0.6102z^{-1}}$$

8.11　IIR 滤波器设计程序的应用

无论我们采用什么方法，很显然双线性变换法包含大量的代数运算，很容易出现错误。利用双线性法来计算滤波器系数的计算机程序包含在文献中或商业软件中，它只需指定感兴趣的滤波器的参数（IEEE, 1979; Gray and Markel, 1976; Parks and Burrus,1987; Jong, 1982; DeFatta et al., 1988）。文献中的大多数程序是用 FORTRAN 编写的。从这个语言到更现代的语言（例如 C 或者 BASIC）需要做一些转换。在本书指导手册的 CD 上，有一个利用 BZT 来计算滤波器系数的 C 语言程序（详情请参见前言）。在下面的例子里将解释这个程序的应用。在附录8B 里我们将说明如何利用 MATLAB 来设计多种 IIR 数字滤波器。

例 8.18　IIR 设计程序应用举例　求一个音频数字滤波器的系数，该滤波器具有切比雪夫特性，且满足下面的性能规范：

通带	0 ~ 2.5 kHz
阻带边沿	2820 kHz
通带波纹	0.47 dB
抽样频率	10 kHz
滤波器阶数	4

解：

利用该程序，得到下面的列表：

k	A_k	B_k
0	$1.000\,000 \times 10^{0}$	$1.934\,410 \times 10^{-1}$
1	$-2.516\,884 \times 10^{-1}$	$3.783\,311 \times 10^{-1}$
2	$1.054\,118 \times 10^{0}$	$5.241\,429 \times 10^{-1}$
3	$-2.406\,030 \times 10^{-1}$	$3.783\,311 \times 10^{-1}$
4	$1.985\,861 \times 10^{-1}$	$1.934\,410 \times 10^{-1}$

8.12　IIR 滤波器的系数计算方法的选择

对于冲激不变法，在对模拟滤波器进行数字化后，原始模拟滤波器的冲激保持不变，但它的幅度-频率响应发生了变化。因为固有的混叠，这个方法对高通或者带阻滤波器是不合适的。另一方面，双线性 z 变换法，可生成很有效的滤波器，非常适合于频率选择滤波器系数的计算。它允许对具有经典特性（例如巴特沃斯、切比雪夫和椭圆特性）的滤波器进行数字滤波器的设计。从双线性变换法得到的数字滤波器，一般来说保持了模拟滤波器的幅度响应特性的某些特征（比如，带沿频率、通带波纹和阻带衰减），但不是必需的时域特性。冲激不变法对模拟具有低通特性的模拟系统是比较好的，但是双线性法对于频率选择 IIR 滤波器是最好的。匹配 z 变换法和冲激不变法有许多共同的固有问题。对于简单的滤波器应用，极-零点放置法提供了简单但有效的获得滤波器系数的方法。

8.12.1　奈奎斯特效应

把模拟滤波器转化成等价离散时间滤波器的三种方法（即匹配 z 变换法、冲激不变法和双线性 z 变换法），在特定的情况下都对滤波器特性（比如幅度、相位和群延迟响应）有着显著的影响。如前面陈述过的，模拟滤波器可用的频带从零一直扩展到无穷，然而对数字滤波器它就只能

从零到奈奎斯特频率（也即是抽样频率的一半）。因此，利用上述三种方法任意一种设计的数字滤波器，它的幅度-频率响应可能和模拟滤波器有着显著的差别，因为整个模拟频带（零到无穷）现在压缩到了一个窄的频带（零到奈奎斯特频率）。这个差别代表了一种失真，有时也称之为奈奎斯特效应。

在许多应用中，除了要预防比规定更大的衰减之外，奈奎斯特效应并不是有害的。然而，在希望保持模拟滤波器响应的应用中，例如在专业和半专业的音频工作中（Clark et al., 1996; 2000），这个影响代表不希望的失真。在这种应用中，选择哪种方法将模拟滤波器转化为等价的离散时间的滤波器，失真程度将是要考虑的一个因素。对其他滤波器特性的影响，例如群延迟和冲激响应，也可能是方法选择时要考虑的一个因素（Rabiner and Gold, 1975）。

在本节里，我们将简要地考虑奈奎斯特效应的后果。在本章后面给出了许多习题，允许读者对这些把模拟滤波器转换成等价离散滤波器的方法的相关优势进行探索。

例8.19 设计一个低通离散时间的滤波器，滤波器具有巴特沃斯特性，要求满足下面的性能规范：

截止频率	300 Hz
滤波器阶数	5
抽样频率	1000 Hz

(1) 在MATLAB的帮助下，

(a) 利用冲激不变法求并且画出滤波器的幅度-频率响应和群延迟响应；

(b) 利用双线性z变换法求并且画出幅度-频率响应和群延迟响应。

(2) 根据由奈奎斯特效应引起的幅度响应失真，对两种方法（双线性z变换和冲激不变法）进行比较。

解：

(1) (a) 我们先来设计模拟滤波器作为基准点。在程序8.1里列出了MATLAB的m文件。通过这个程序得到的滤波器的幅度-频率响应，绘制在图8.18中。

在图8.19(a)和图8.19(b)里给出了等价离散滤波器的幅度-频率响应和群延迟响应，该滤波器是通过冲激不变法得到的。滤波器的MATLAB m文件在程序8.2中给出。利用MATLAB信号处理工具箱的grpdelay得到群延迟响应。

程序8.1 模拟滤波器设计的MATLAB m文件

```
%
% Program name: EX8-1.m
%
FN=1000/2;
fc=300;                              % Cutoff frequency
N=5;                                 % Filter order
[z, p, k]=buttap(N);                 % Create an analog filter
w=linspace(0, FN/fc, 1000);          % Plot the response of filter
h=freqs(k*poly(z), poly(p), w);
f=fc*w;
plot(f, 20*log10(abs(h))), grid
ylabel('Magnitude (dB)')
xlabel('Frequency (Hz)')
```

程序 8.2　冲激不变滤波器设计的 MATLAB m 文件

```
%
%   Program name: EX8-2.m
%
Fs=1000;                          %   Sampling frequency
fc=300;                           %   Cutoff frequency
WC=2*pi*fc;                       %   Cutoff frequency in radian
N=5;
[b,a]=butter(N, WC, 's');         %   Create an analog filter
[z,p,k]=butter(N, WC, 's');
[bz, az]=impinvar(b, a, Fs);      %   Determine coeffs of IIR filter
[h, f]=freqz(bz, az, 512, Fs);
plot(f, 20*log10(abs(h))), grid
xlabel('Frequency (Hz)')
ylabel('Magnitude Response (dB)')
```

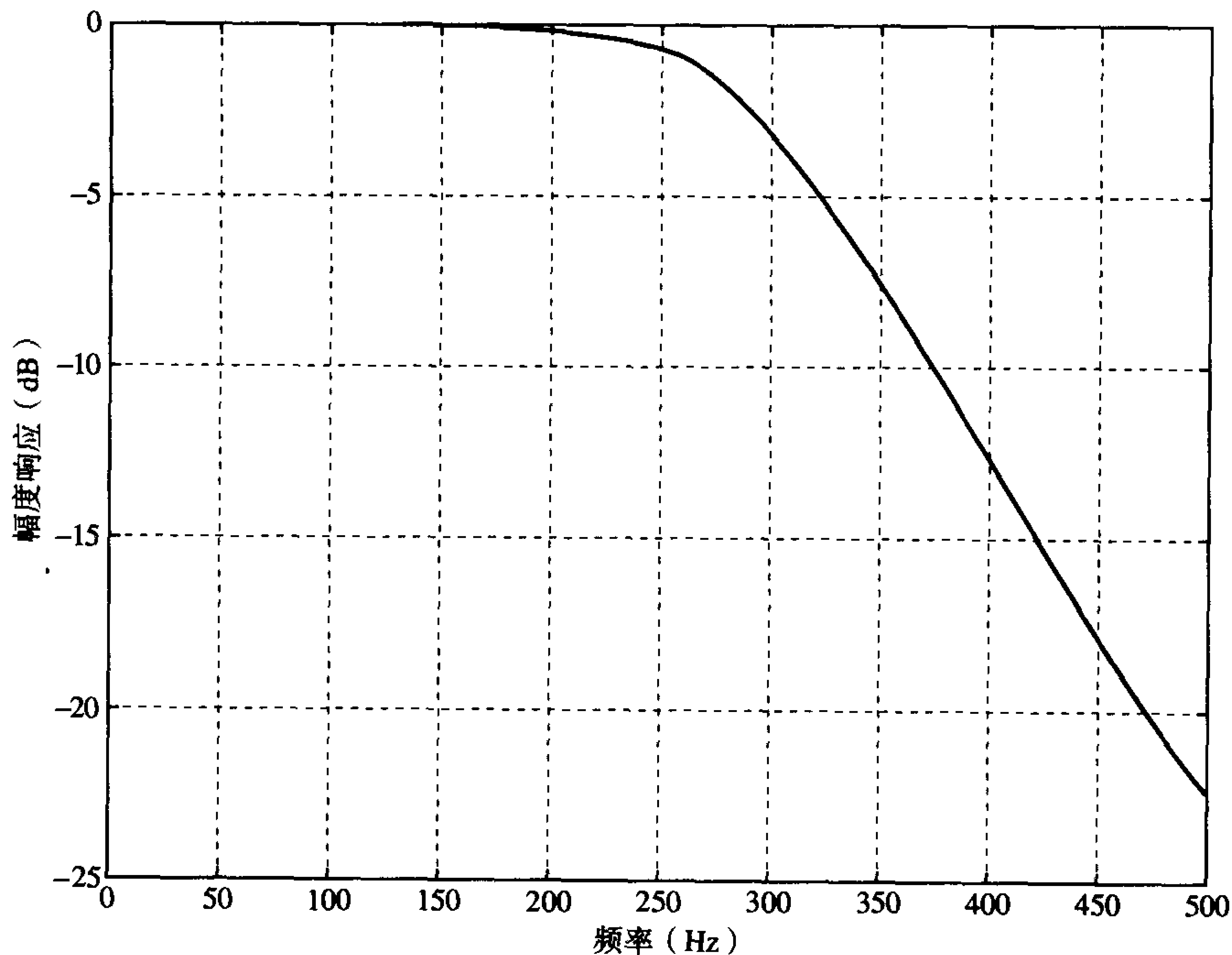
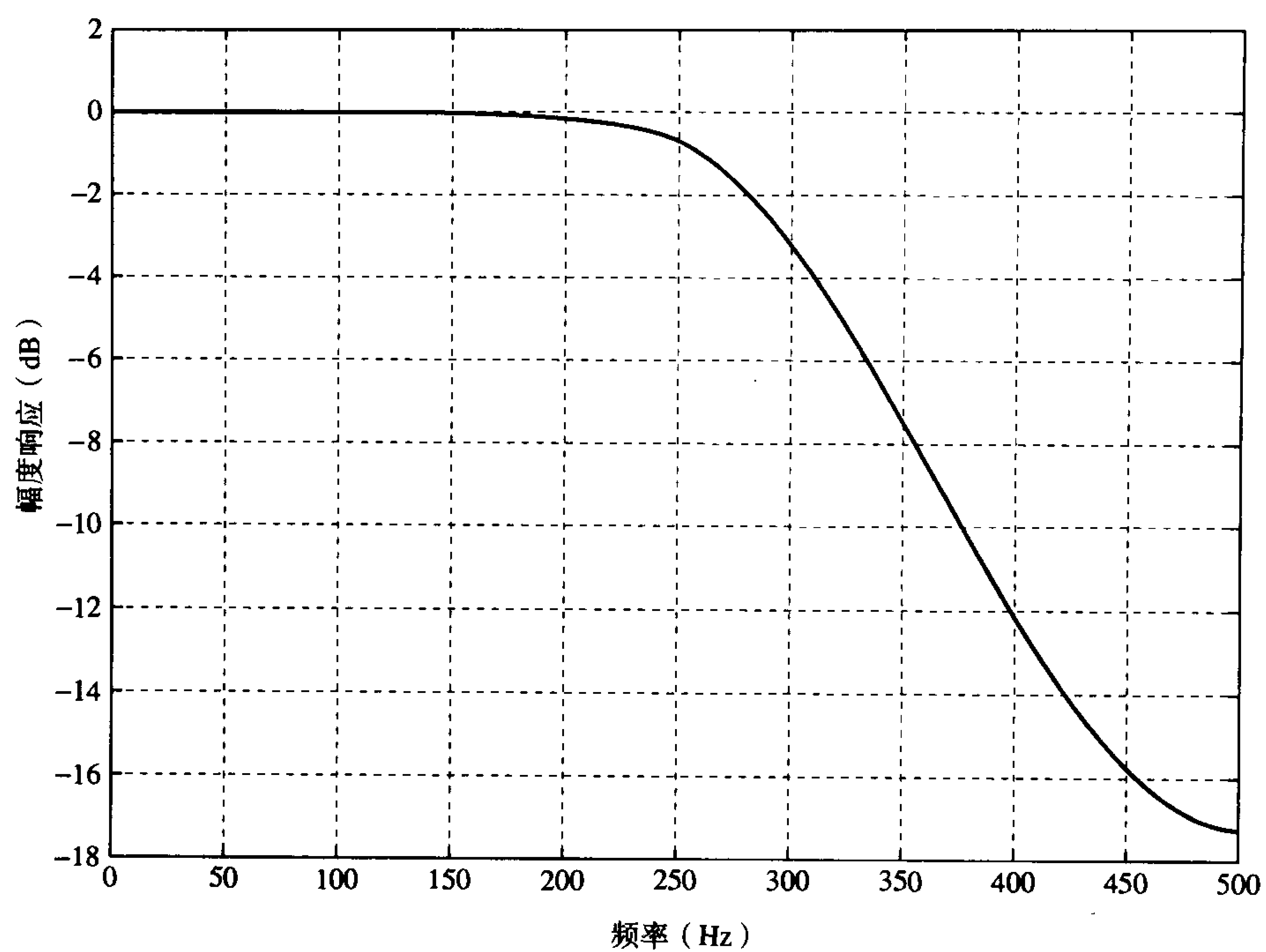


图 8.18　模拟滤波器的幅度响应

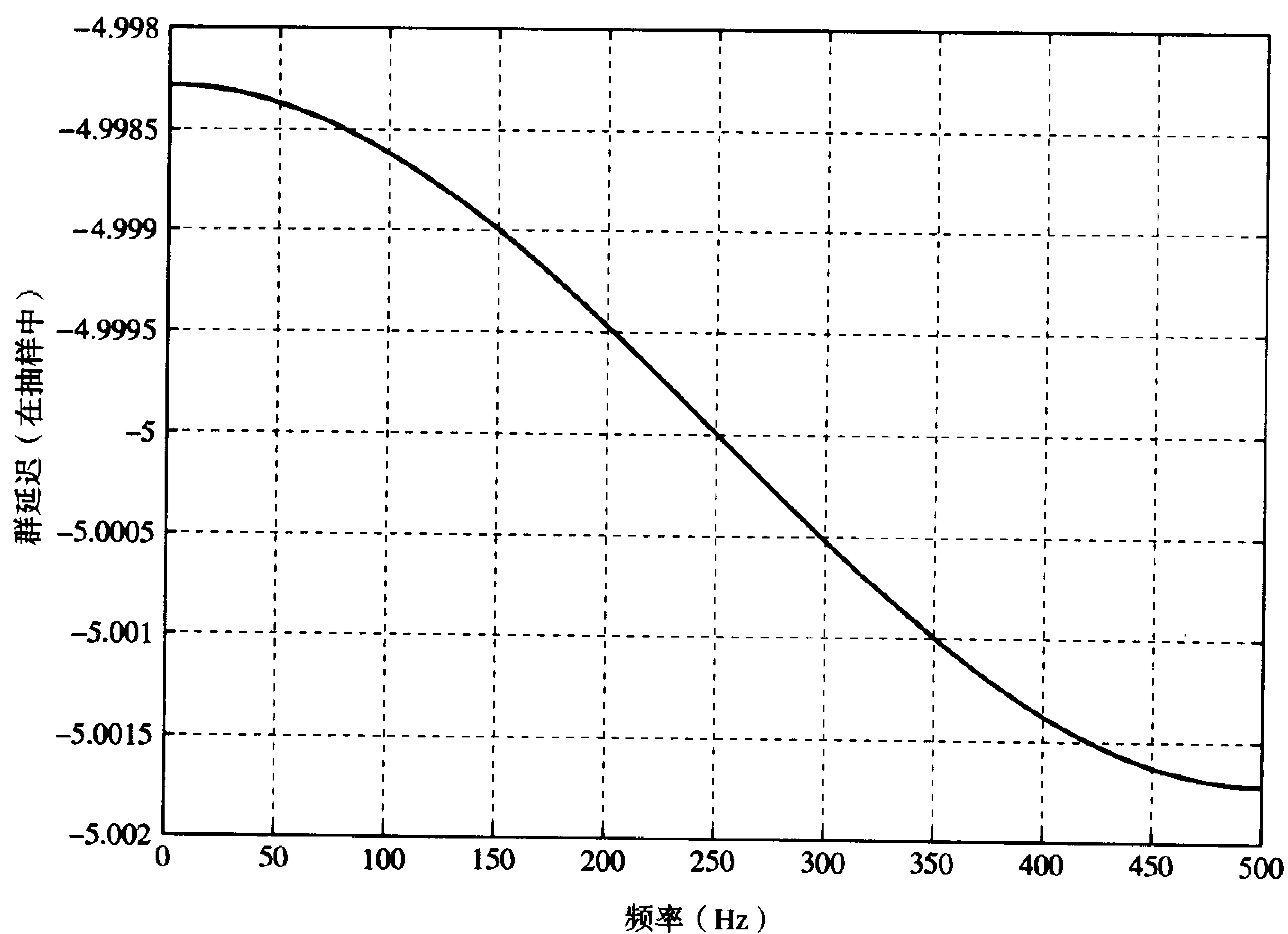
(b) 通过BZT法设计的等价离散滤波器的幅度-频率响应和群延迟响应，如图8.20(a)和图8.20(b)所示。滤波器的 MATLAB m 文件如程序 8.3 所示。

程序 8.3　BZT 滤波器设计的 MATLAB m 文件

```
%
%   Program name: EX8-3.m
%
Fs=1000;                          %   Sampling frequency
FN=Fs/2;
fc=300;                           %   Cutoff frequency
N=5;
[z, p, k]=butter(N, fc/FN);
[h, f]=freqz(k*poly(z), poly(p), 512, Fs);
plot(f, 20*log10(abs(h))), grid
ylabel('Magnitude (dB)')
xlabel('Frequency (Hz)')
```

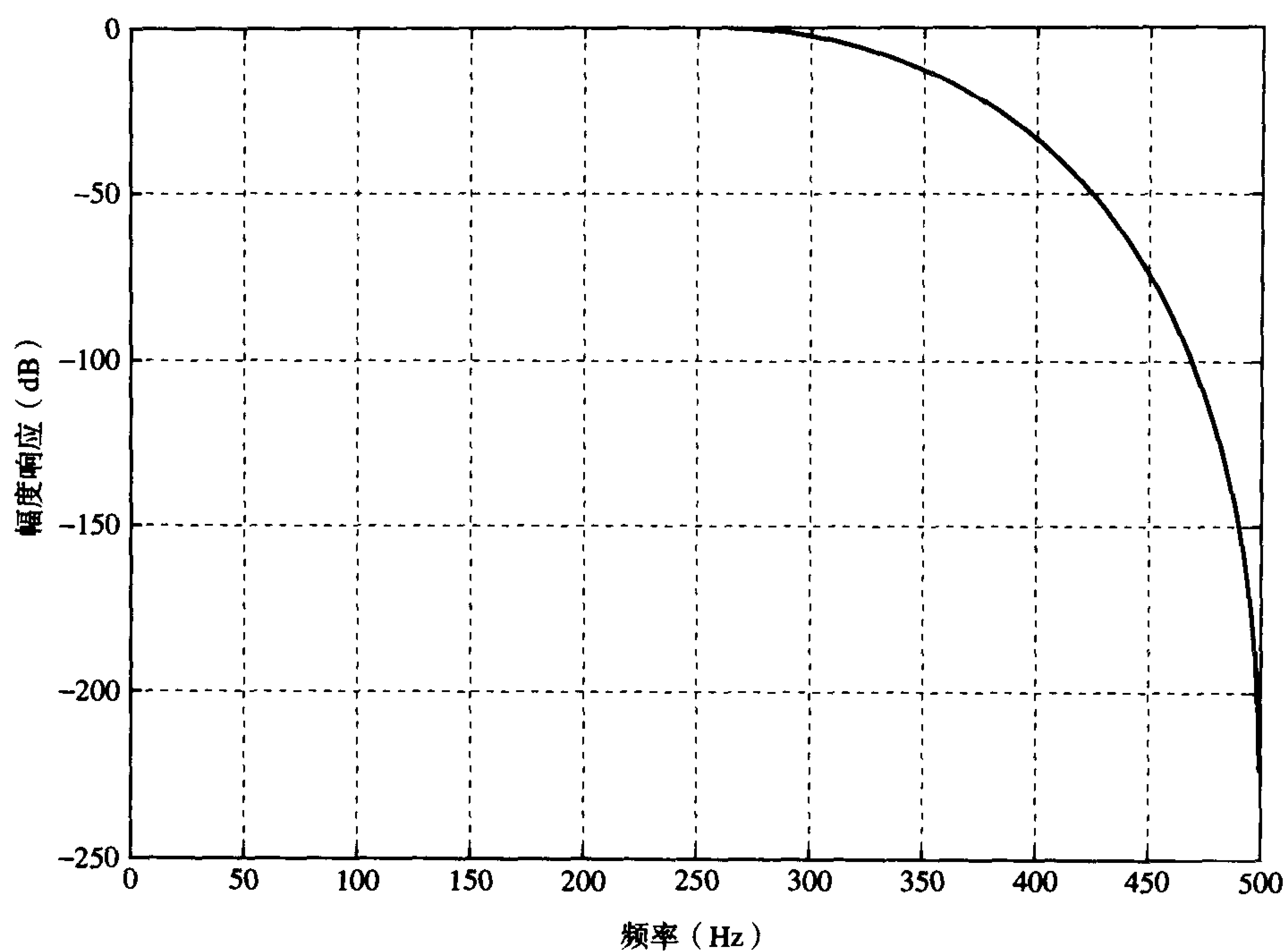


(a) 冲激不变法

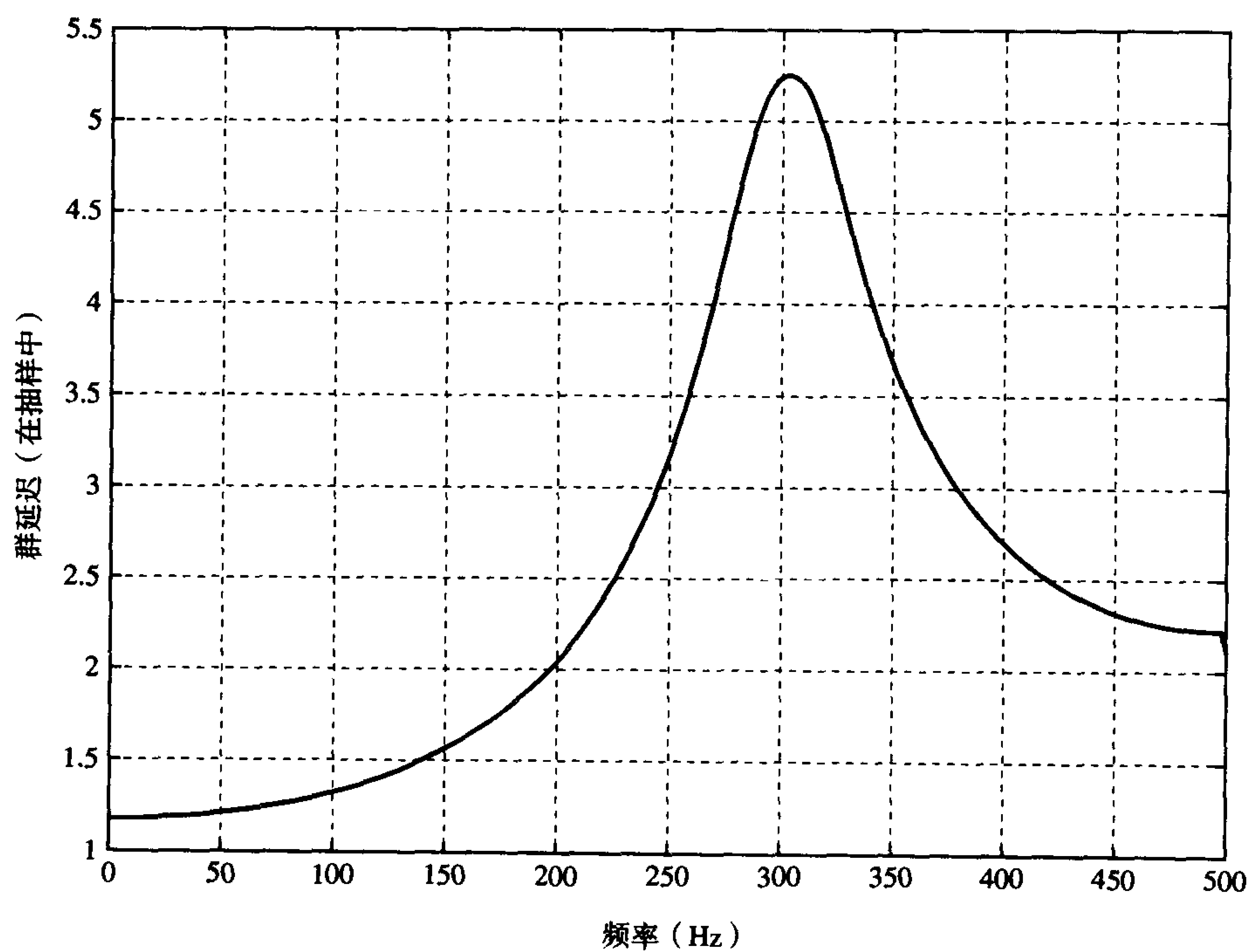


(b) 冲激不变法的群延迟响应

图 8.19 等价离散滤波器（利用冲激不变法）的幅度－频率响应和群延迟响应

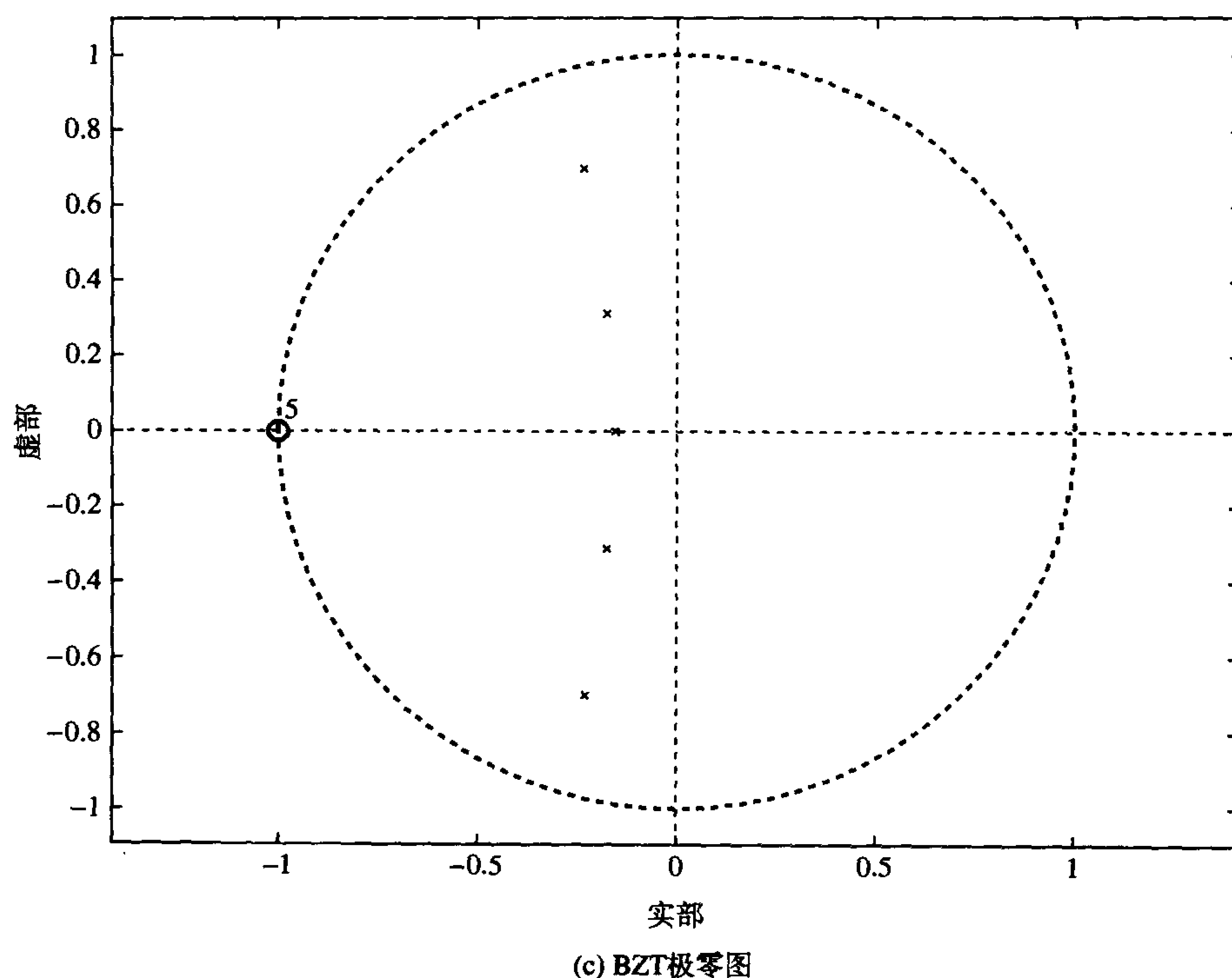


(a) BZT的幅度-频率响应



(b) BZT群延迟响应的例子

图 8.20　等价离散滤波器（利用 BZT）的幅度 - 频率响应和群延迟响应及极零图



(c) BZT极零图

图 8.20（续） 等价离散滤波器（利用BZT）的幅度－频率响应和群延迟响应及极零图

(2) 如果我们把冲激不变滤波器和双线性 z 变换滤波器的幅度－频率响应和模拟滤波器的相比较，很显然在奈奎斯特频率附近，冲激不变滤波器提供的衰减较少，而BZT滤波器的衰减较多。例如，在500 Hz，冲激不变滤波器的衰减大约是17 dB，而BZT滤波器的衰减超过了200 dB。

对BZT滤波器进行分析可以看出，传递函数在奈奎斯特频率处有五个零点（参见图8.20(c)），这是造成幅度响应急剧下降的原因。这在基于BZT的离散滤波器中是很常见的，也是为什么它的衰减比原始的模拟滤波器大的原因。与此相反，基于冲激不变法的离散滤波器在原点和单位圆以外各有一个零点。这两种离散滤波器和模拟滤波器的幅度响应的差别表现为一种失真。

在许多情况下，MZT（或者冲激不变）和BZT的幅度响应的失真有相反的效应。基于BZT的滤波器趋向于提供较大的衰减，而MZT和基于冲激不变的滤波器提供比较小的失真。在低频和中频带（相对于奈奎斯特频率而言），通过MZT和冲激不变法设计的低通和带通滤波器的幅度响应，相当接近原始模拟滤波器的幅度响应，但是在奈奎斯特频率附近就会变坏。通过BZT法设计的滤波器的幅度响应也是如此，除了在接近奈奎斯特频率处的响应变差是一种相反的情况以外。对于低通和带通滤波器，防止响应失真的一种简单而有效的方法是把BZT法与MZT法或者冲激不变法通过对它们的系数取平均而组合在一起（Clark et al., 1996; 2000）。在这种方式里，低通和带通滤波器首先是独立用BZT和MZT来设计的。接着对系数取平均得到如下的一组新的系数：

$$b'_k = [b_k(BZT) + b_k(MZT)]/2,\ k = 0, 1, 2$$

$$a'_k = [a_k(BZT) + a_k(MZT)]/2,\ k = 0, 1, 2$$

图8.21比较了两个离散滤波器的响应，一个是通过MZT法设计的，另一个是通过BZT法设计的。这个滤波器是打算在音频信号处理应用中使用的，它需要一个有效的手段来在线生成滤波

器系数，且响应失真小。如同前面一样，基于 BZT 的滤波器在超过 10 kHz（奈奎斯特频率是 24 kHz）时产生较锐利的幅度 - 频率响应，而 MZT 产生的衰减较小。幅度响应的两个偏移代表了在这个应用中的失真。另一方面，平均后的幅度响应（参见图 8.21）减少了响应失真。图 8.22 的极零图比较了 MZT 和 MZT/BZT 平均滤波器的极点和零点的位置。如我们期望的那样，平均处理改变了 MZT 滤波器的极点和零点的位置，特别是它在奈奎斯特频率处产生了一个“软”零点。这个零点称为“软”的，是因为它位于单位圆的最里面，在 BZT 里不会造成滤波器的幅度响应迅速跌到零。

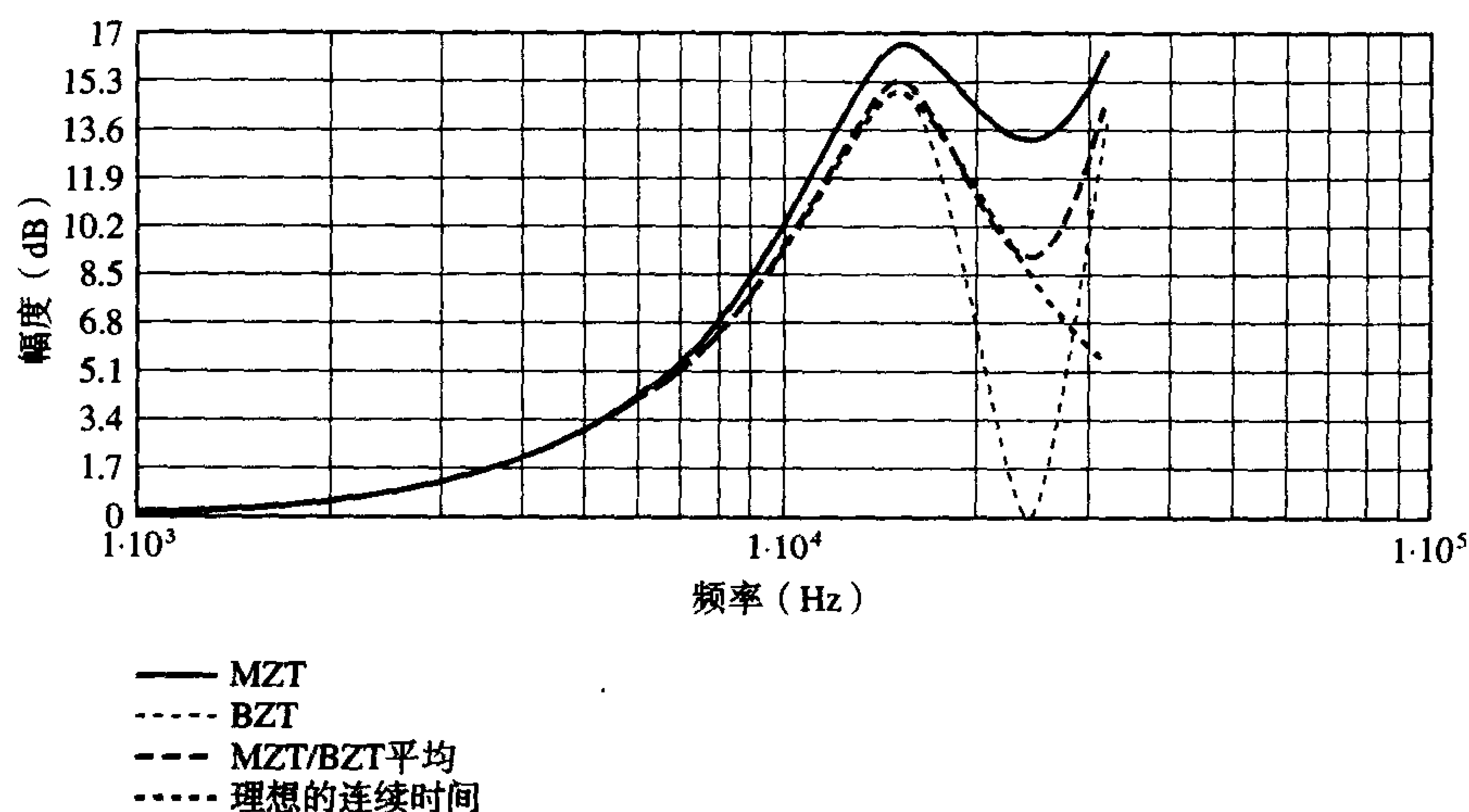


图 8.21　MZT、BZT、MZT/BZT 平均的幅度响应，以及理想的连续时间钟形滤波器的幅度响应，钟形滤波器在频率 15 kHz 处，$Q = 2$，15 dB 提升

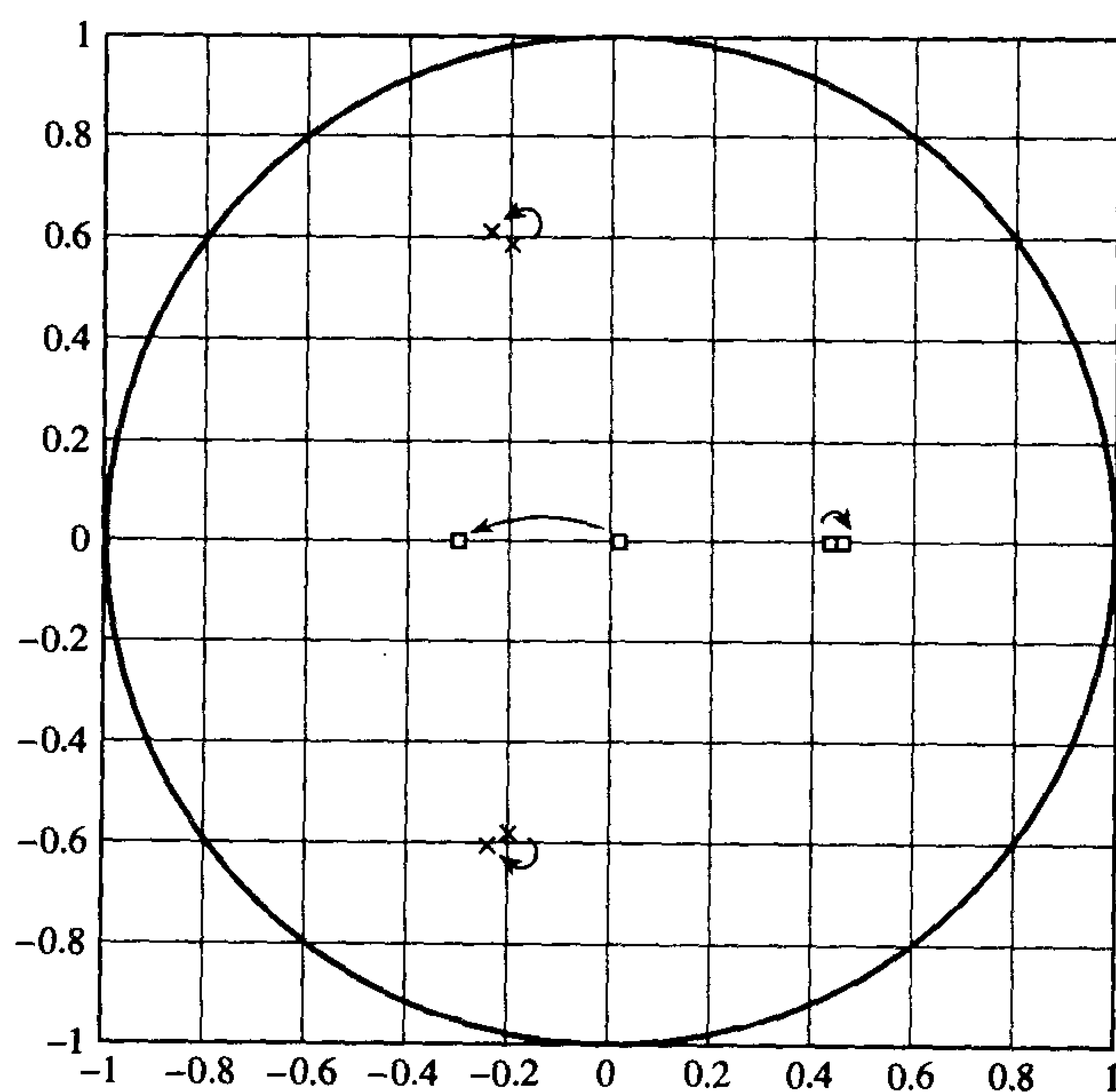


图 8.22　MZT 和 MZT/BZT 平均极点和零点的比较图。箭头指出了极 - 零点从 MZT 到 MZT/BZT 平均的移动

8.13 IIR数字滤波器的实现结构

实现结构就是把给定的传递函数$H(z)$转化成相匹配的滤波器结构。一般用流程图或框图来描述滤波器结构，它们表示了实现数字滤波器的计算过程。实现结构的基本单元是乘法器、加法器和延迟单元；参见图8.23。

回顾一下由下式刻画的IIR滤波器：

$$H(z)=\sum_{k=0}^{N}b_k z^{-k}\bigg/\left(1+\sum_{k=1}^{M}a_k z^{-k}\right)\quad M\geqslant N \tag{8.46a}$$

$$y(n)=\sum_{k=0}^{N-1}b_k x(n-k)-\sum_{k=1}^{M}a_k y(n-k) \tag{8.46b}$$

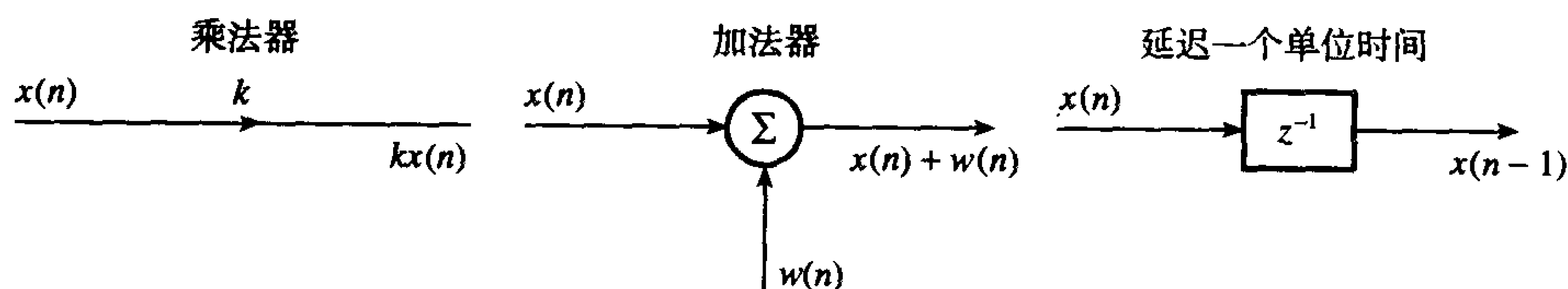


图8.23 滤波器结构的基本单元

图8.24给出了7.46式所示的直接实现形式，其中为了简化令$N=M$。注意在图表中用到的系数和传递函数中用到的是相同的，但是和分母系数的符号是相反的。当滤波器阶数变高时，例如$M>3$，图8.24所示的滤波器的直接实现形式对有限字长效应是非常敏感的，在这种情况下是要避免用这种直接形式的。在实际中，$H(z)$通常被分解为小的部分，典型的是二阶和/或者一阶构建块，然后再串联或者并联起来（参见后面）。

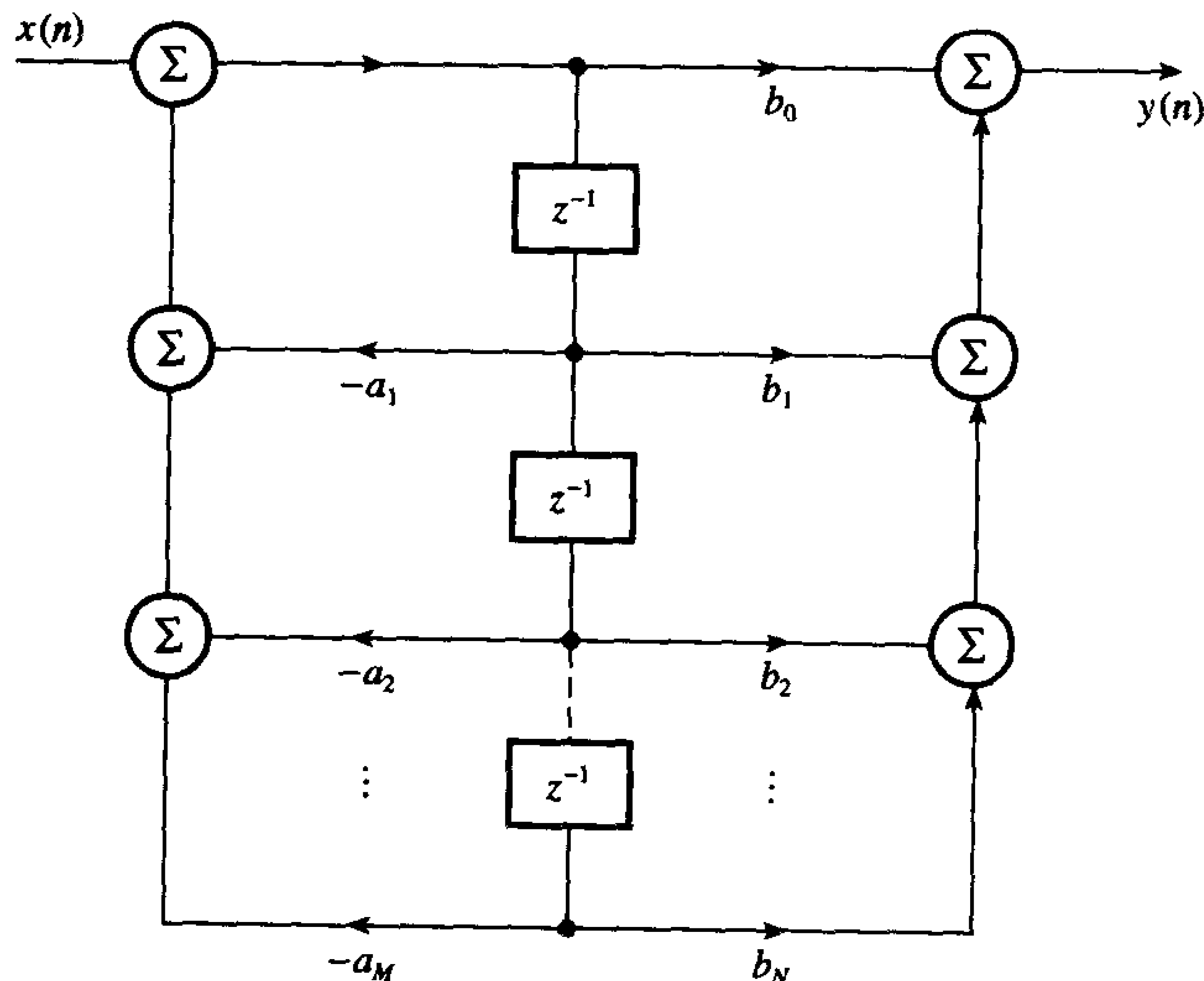


图8.24 IIR滤波器的直接实现形式

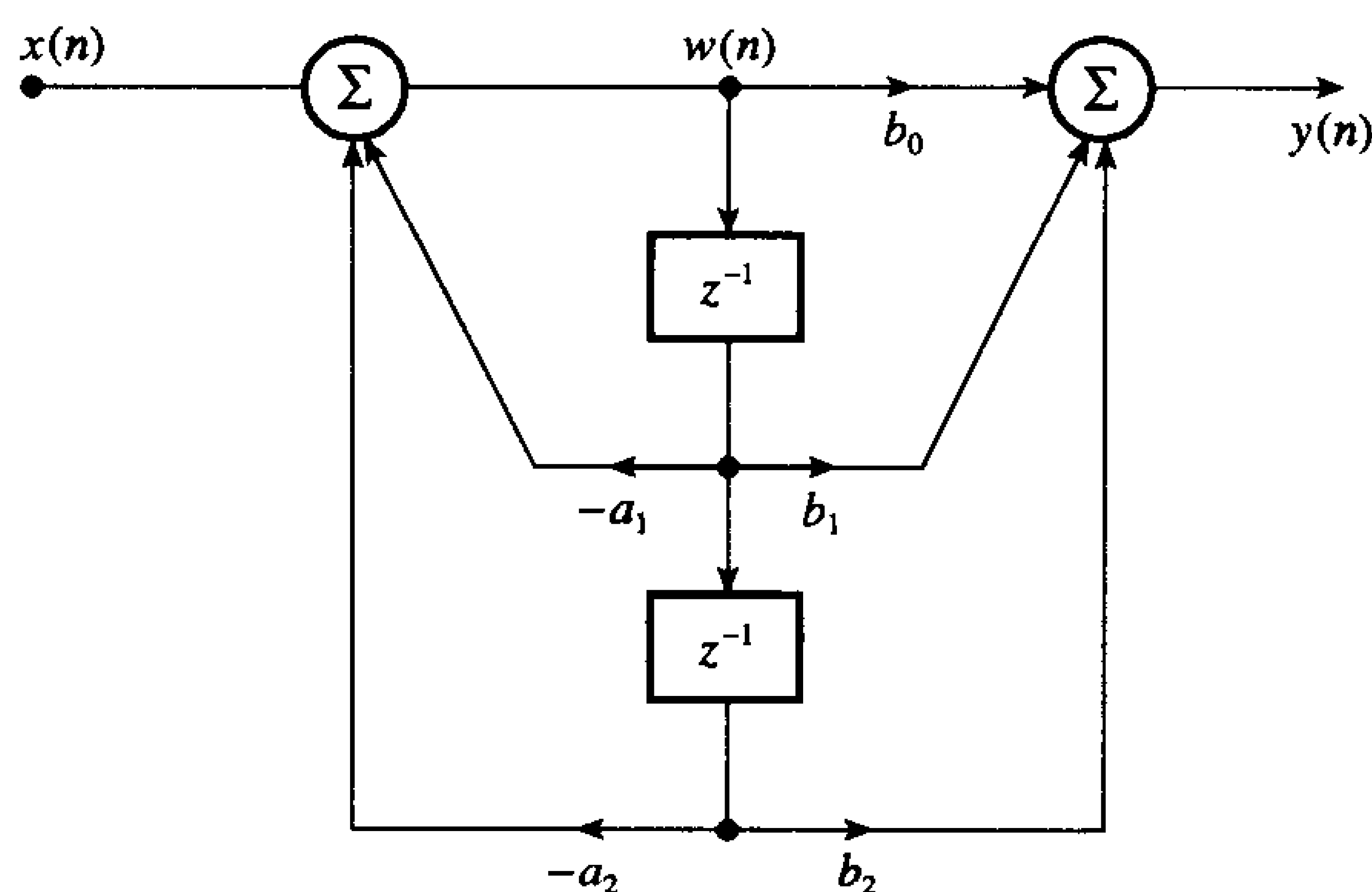
8.13.1 IIR滤波器的实际构件

图8.25描绘了在实践中实现高阶IIR滤波器时用到的二阶构件。第一个（参见图8.25(a)）经常称为标准单元（或者直接形式2），因为它的延迟单元的数目最少。这种双二次型部分，它的特性由下面的方程刻画：

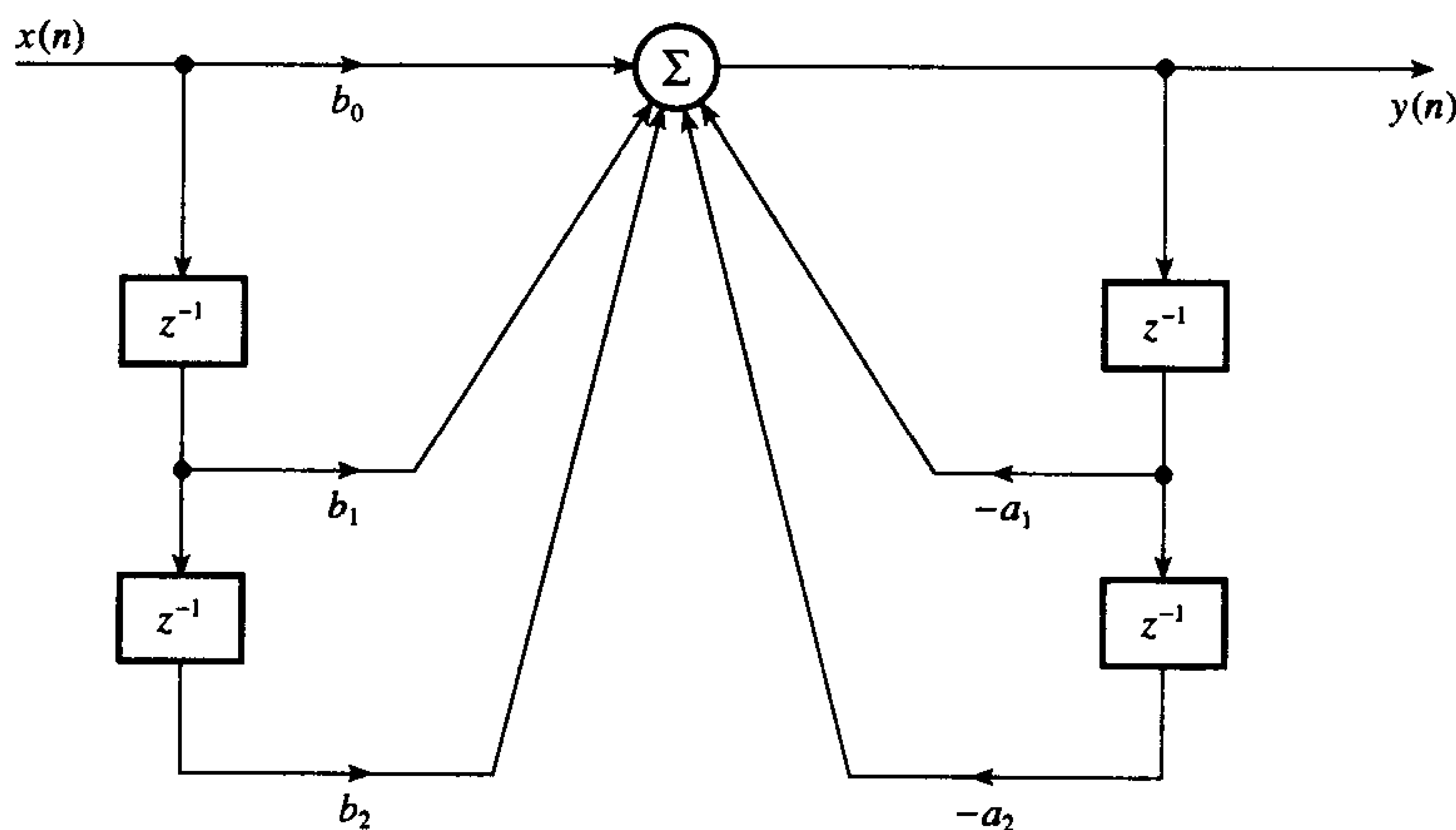
$$w(n)=x(n)-\sum_{k=1}^{2}a_k w(n-k) \tag{8.47a}$$

$$y(n)=\sum_{k=0}^{2}b_k w(n-k) \tag{8.47b}$$

$$H(z)=\frac{b_0+b_1z^{-1}+b_2z^{-2}}{1+a_1z^{-1}+a_2z^{-2}} \tag{8.47c}$$



(a) 标准二阶部分



(b) 二阶部分的直接形式

图 8.25　IIR 滤波器实现结构的实际构造框图

第二个滤波器部分（参见图 7.18 (b)）是二阶 IIR 方程的直接实现结构。它的特性是由下面的方程来刻画：

$$y(n)=\sum_{k=0}^{2}b_k x(n-k)-\sum_{k=1}^{2}a_k y(n-k) \tag{8.48a}$$

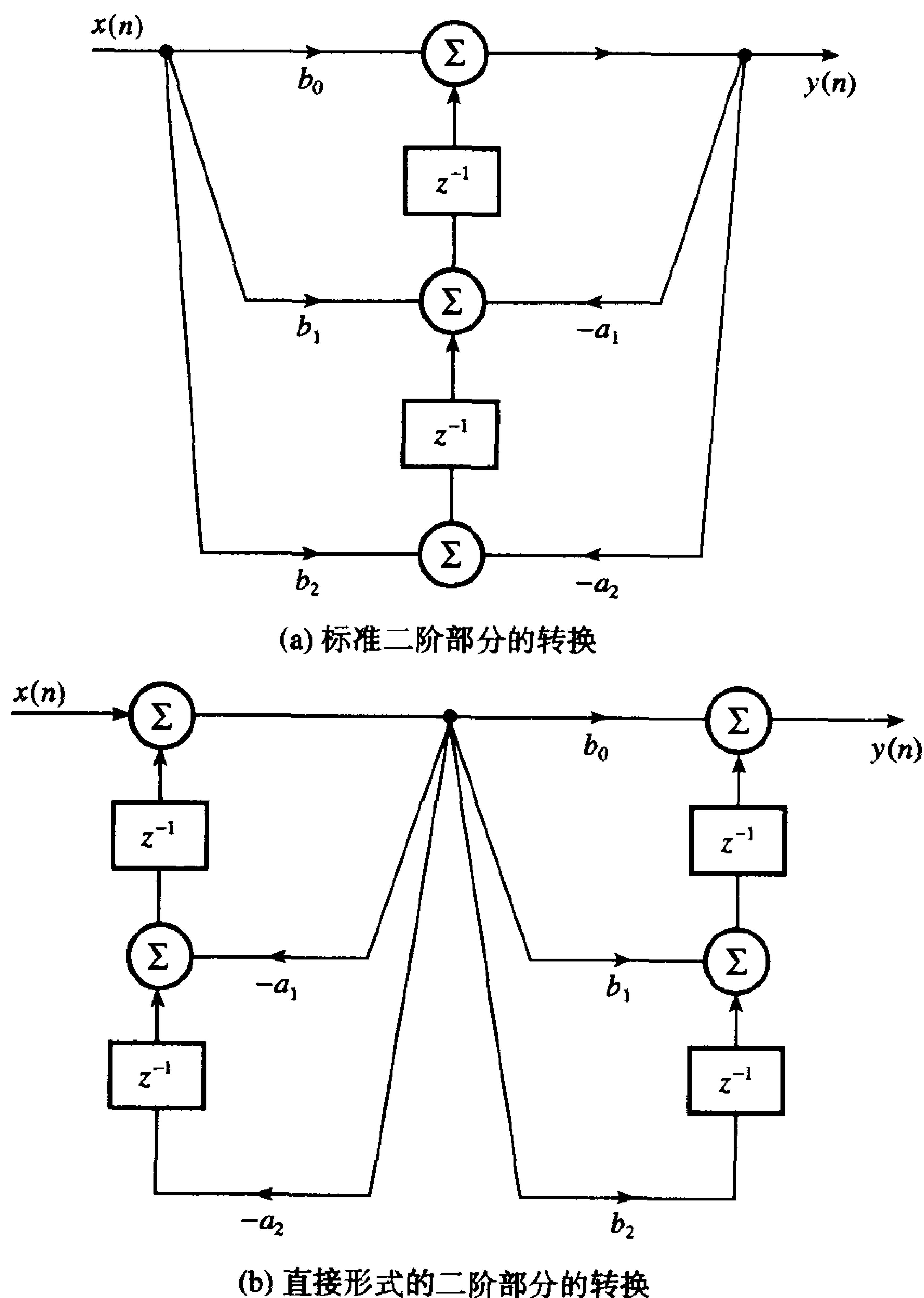
$$H(z)=\frac{b_0+b_1z^{-1}+b_2z^{-2}}{1+a_1z^{-1}+a_2z^{-2}} \tag{8.48b}$$

标准部分（参见图 8.25(a)）是最流行的，因为它有好的舍入噪声特性，并且要求的存储单元的数目最少，但是它对内部的溢出非常敏感。为了避免内在溢出，需要对滤波器单元的输入进行伸缩变换。对于直接型（参见图 8.25(b)），伸缩不是一定要求的，因为它仅有一个加法器，当不希望进行伸缩时，可能它是最好的，例如高保真的数字音频（Dattorro, 1988）。在某些情况下，从噪声性能来说，直接型要优于标准部分。

耦合形式具有一些我们希望的有限字长特性（Gold and Rader, 1969），但是它要求的计算量更大，不容易用于实现具有二阶分子系数的传递函数。

图 8.25 给出的滤波器块一般是二阶部分。从它们也可以推导出其他的滤波器块。例如，如果图 8.25(a)里的分子系数 a_1 和 a_2 都是零，那么我们有一个纯粹的递归结构。另一方面，如果滤波器参数是用椭圆函数得到的，那么系数 a_2 是 1。在上述的任意结构里，通过令 $a_2 = b_2 = 0$，很容易得到一阶滤波器块。

图 8.26 给出了二阶标准滤波器单元和直接形式的滤波器单元的转换。它们分别是从图 8.26(a)和图 8.26(b)出发，交换所有的加法器和分支节点并颠倒箭头的方向得到的。尽管图 8.26 里的传递函数和它们转换后的相同，但是它们的有限字长特性是相当不同的。其他对有限字长效用较不敏感的结构也存在，但是通常来说它们复杂得多，例如有最小噪声结构、状态变量结构和格型结构等。



(a) 标准二阶部分的转换

(b) 直接形式的二阶部分的转换

图 8.26 二阶标准滤波器单元和直接形式的滤波器单元的转换

8.13.2 高阶 IIR 滤波器的串联和并联实现结构

在实际应用中，高阶传递函数是用上述二阶和/或一阶构件串联或者并联实现的。典型的情况是，在串联实现结构中，传递函数被分解成 $N/2$ 个二阶因式：

$$H(z) = \prod_{k=1}^{N/2} \left[\frac{b_{0k} + b_{1k}z^{-1} + b_{2k}z^{-2}}{1 + a_{1k}z^{-1} + a_{2k}z^{-2}} \right] = \prod_{k=1}^{N/2} \frac{N_k(z)}{D_k(z)} \tag{8.49a}$$

其中

$$N_k(z) = b_{0k} + b_{1k}z^{-1} + b_{2k}z^{-2}$$
$$D_k(z) = 1 + a_{1k}z^{-1} + a_{2k}z^{-2} \tag{8.49b}$$

N为滤波器的阶数，它假设为偶数。如果N是奇数，那么$H_k(z)$中的某一个将是一阶部分。

每一个二阶因式$H_k(z)$可以利用一个构件来实现，这些构件串联在一起；参见图 8.27。在串联实现结构中出现了 3 个困难：(1)分子因式和分母因式如何组对；(2)应该连接的单个单元的阶数；(3)需要针对滤波器内的不同点对信号电平进行伸缩变换，避免电平变的太大或太小。

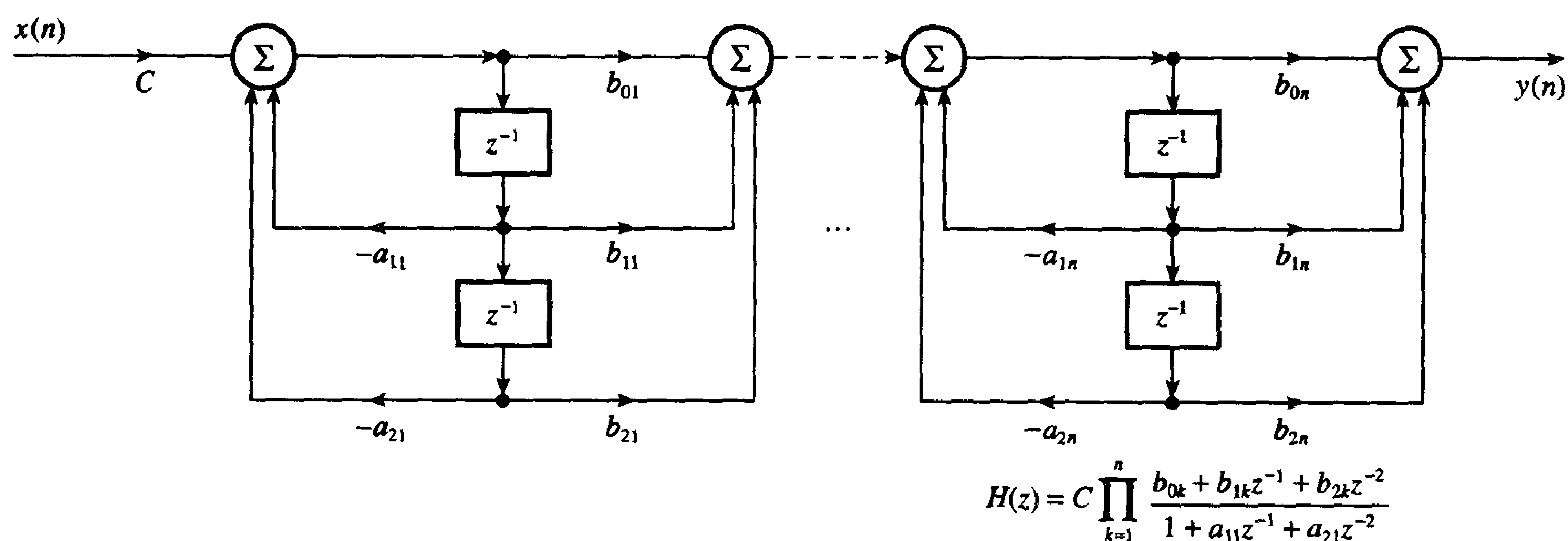


图 8.27　串联实现

分子因式和分母因式可以按许多方法来整序。例如，四阶滤波器可以分解成两个二阶部分，接着以四种不同方法中的某一个来组对和整序：

(1)　$H(z) = \dfrac{N_1(z)}{D_1(z)} \dfrac{N_2(z)}{D_2(z)}$

(2)　$H(z) = \dfrac{N_2(z)}{D_2(z)} \dfrac{N_1(z)}{D_1(z)}$

(3)　$H(z) = \dfrac{N_1(z)}{D_2(z)} \dfrac{N_2(z)}{D_1(z)}$

(4)　$H(z) = \dfrac{N_2(z)}{D_1(z)} \dfrac{N_1(z)}{D_2(z)}$

其中每一个$N_k(z)$和$D_k(z)$都是8.49b式中定义的二阶多项式。在第一种情况下，第一个滤波器部分是由分子和分母对$N_1(z)$和$D_1(z)$组成，第二个滤波器部分是由分子和分母对$N_2(z)$和$D_2(z)$组成。很显然，分子和分母可能的组对和排列组合的数目是相当大的。一般情况下，对于N阶滤波器来说，可能的不同组对和排序的数目是

$$\left(\frac{N}{2}!\right)^2 \tag{8.50}$$

最首要的原则是：如果$N_i(z)$的零点最接近$D_k(z)$的极点，那么就把$N_i(z)$和$D_k(z)$组成一对，这样避免在极点对应的频率处有大的幅度响应；另外把含有最靠近单位圆的极点的二阶部分放在串联的最后面（Jackson, 1986）。对于滤波器部分的组对和排序，已经开发出了许多有效的设计方法，它是基于哪一种排序能得到最好的信噪比。这是一个和组对、排序紧密联系的主题（参见第13章）。

在并联实现结构中，N阶传递函数$H(z)$可以利用部分分式展开成下式：

$$H(z) = C + \sum_{k=1}^{N/2} H_k(z) \tag{8.51}$$

其中

$$C = \frac{b_N}{a_N}, \quad H_k(z) = \frac{b_{0k} + b_{1k}z^{-1}}{1 + a_{1k}z^{-1} + a_{2k}z^{-2}}$$

此外，每一个二阶部分可以用前面描述过的、如图8.28所表示的构件来实现。非常值得注意的是，在并联实现结构中，z^{-2}的分子系数是零。在并联实现结构中，各部分的排序并不重要。此外，伸缩变换是很容易的，每一个模块可以独立地实现它（参见后面）。并联结构的SNR比得上最好的串联实现的SNR（Jackson, 1986）。然而，并联结构的零点对系数量化误差敏感得多。注意，并联结构的零点对系数量化的敏感性，在当系数字长降低到5位或者更少时，似乎最为严重。看起来对于绝大部分滤波器，在系数字长是12位或者更高时，并联和串联结构间的差别很小。不过，当利用BZT从经典模拟滤波器推导出串联模式时，这个串联模式有一个非常重要的优点是25%到50%的滤波器系数为简单的整数（0、±1或者±2）。这对仅具有基本算术能力的系统（这种系统中乘法的数目必须保持很少）非常有吸引力。另外，大多数可用的软件包只计算串联实现结构的系数，而不计算并联结构的系数。因为上述原因串联结构才变得流行起来。

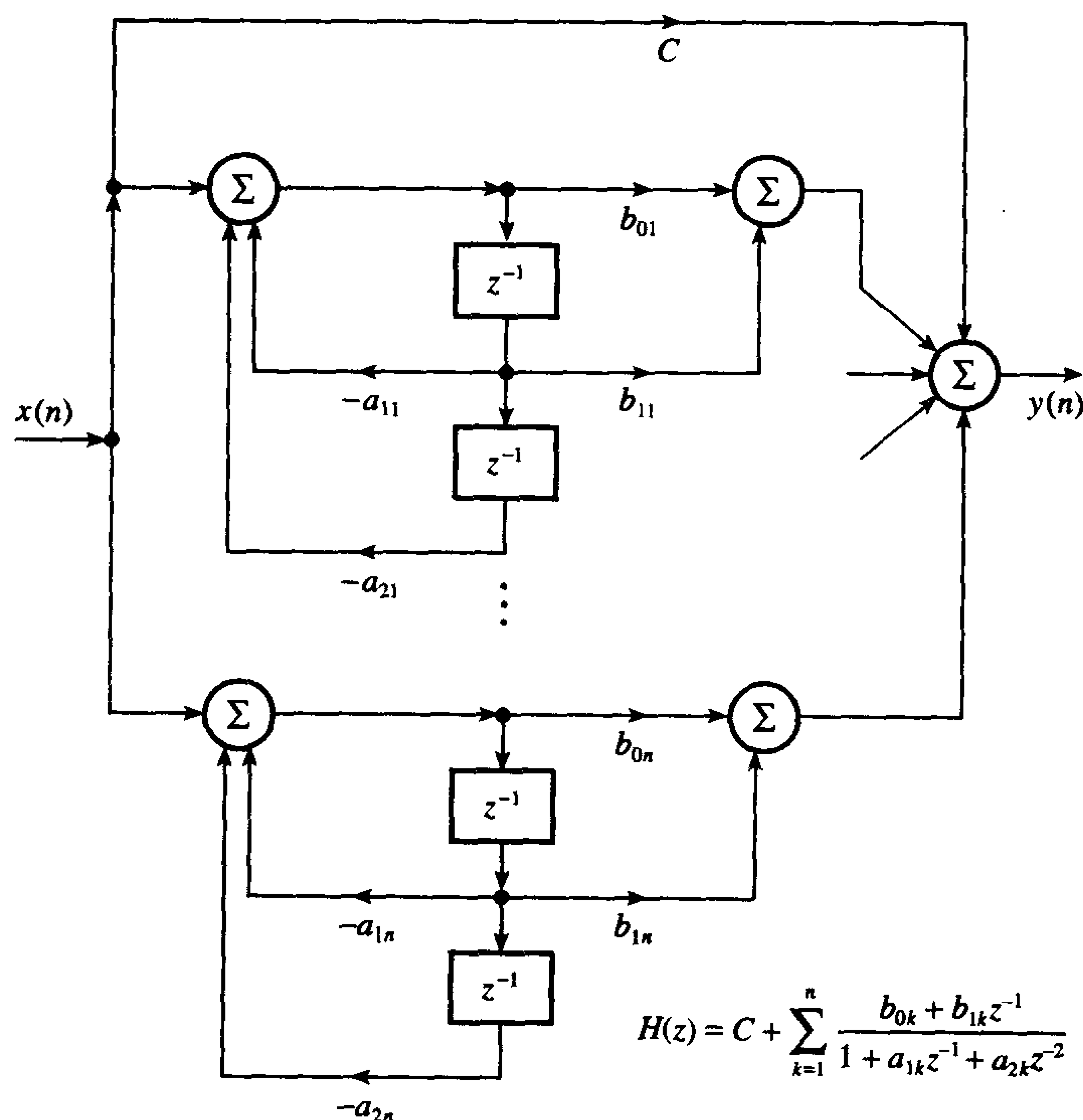


图8.28　并联实现

例 8.20　对于由下列用一阶和二阶传递函数刻画的滤波器，(1)求串联实现结构和(2)求并联实现结构。

$$H(z) = \frac{0.1432(1 + 3z^{-1} + 3z^{-2} + z^{-3})}{1 - 0.1801z^{-1} + 0.3419z^{-2} - 0.0165z^{-3}}$$

解：

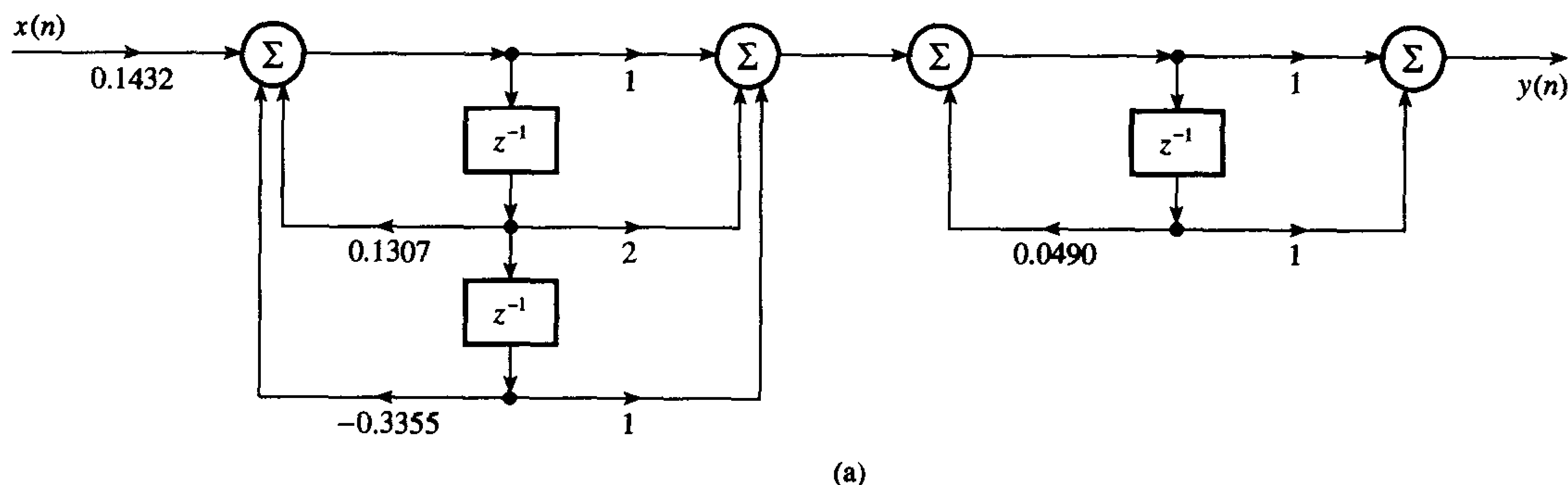
(1) 对于串联实现结构，$H(z)$用因式表示为

$$H(z) = 0.1432 \frac{1 + 2z^{-1} + z^{-2}}{1 - 0.1307z^{-1} + 0.3355z^{-2}} \frac{1 + z^{-1}}{1 - 0.0490z^{-1}}$$

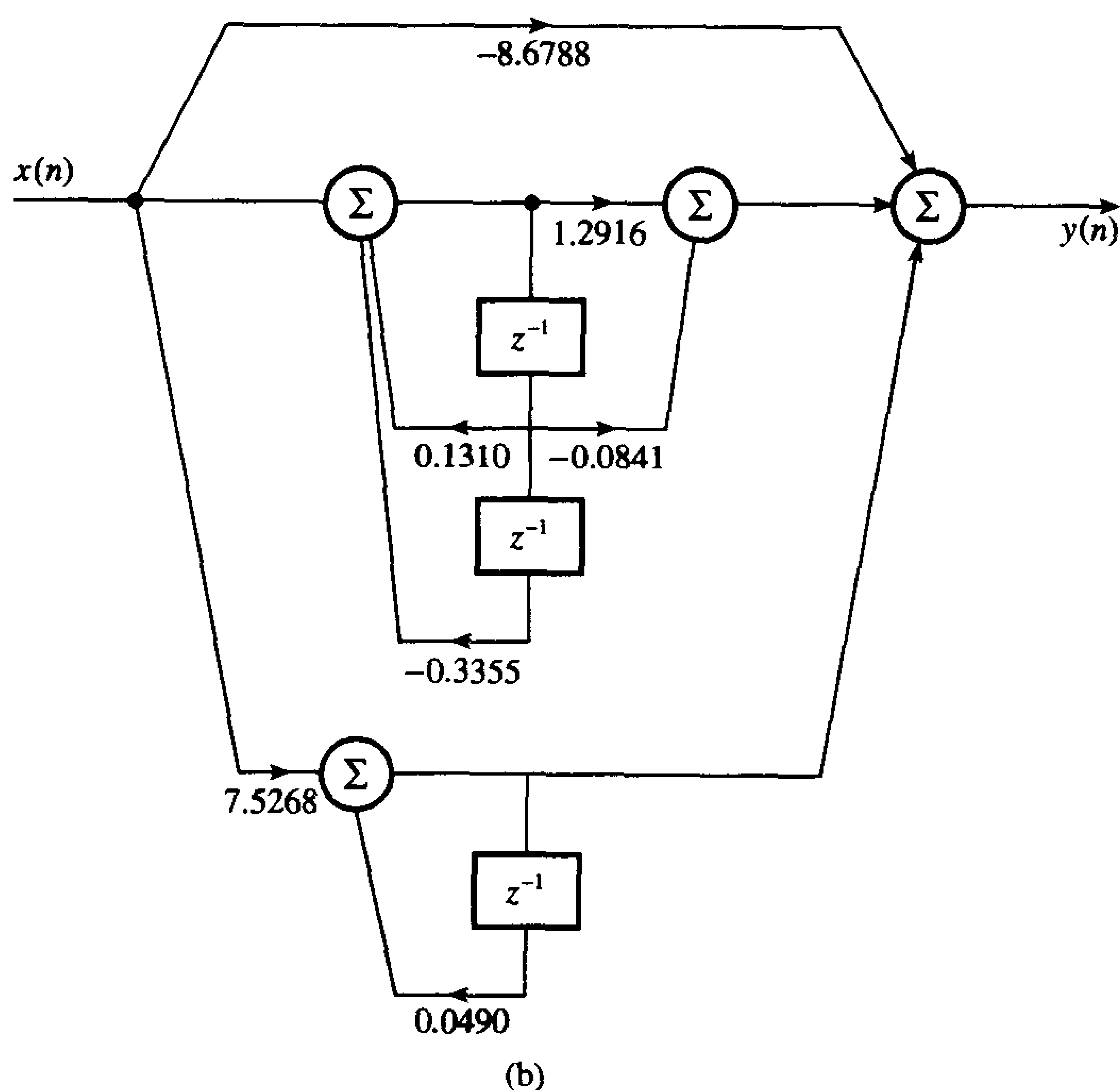
(2) 对于并联实现结构，$H(z)$用因式分解表示为二阶部分和一阶部分的和：

$$H(z) = \frac{1.2916 - 0.08407z^{-1}}{1 - 0.131z^{-1} + 0.3355z^{-2}} + \frac{10.1764}{1 - 0.049z^{-1}} - 8.7107$$

图8.29(a)和图8.29(b)分别给出了串联结构和并联结构的实现图。并联实现的系数是利用第4章中给出的C语言程序得到的。



(a)



(b)

图 8.29　例 8.20 的(a)串联实现结构和(b)并联实现结构

8.14 IIR 滤波器的有限字长效应

前面得到的系数 a_k 和 b_k（参见8.4节～8.10节）是具有无限或者非常高精度的，典型的是6到7位小数位。当IIR数字滤波器是在一个小系统中实现时，例如8位的微计算机中，在表示滤波器系数和执行差分方程中指出的运算时会有误差产生。这些误差降低了滤波器的性能，在极端情况下会导致不稳定。

在实现IIR滤波器之前，确定有限字长效应会在何种程度上降低滤波器的性能，以及如果这种降低程度是不能接受的则要寻找补偿，这是非常重要的。一般来说，这些误差效应可以通过利用更多的位将这种影响降低到可以接受的程度，但这是以提高成本为代价的。

数字 IIR 滤波器的主要误差如下：

- ADC 量化噪声，这是由使用少的位数来表示输入数据 $x(n)$ 的样本引起的；
- 系数量化误差，这是由使用有限位数来表示 IIR 滤波器系数引起的；
- 溢出误差，这是由有限长度寄存器中部分结果的相加或累加引起的；
- 乘积舍入误差，这是由输出结果 $y(n)$ 以及内部算术运算的结果需要四舍五入（或者截尾舍入）到允许的字长而引起的。

滤波器性能的降低程度依赖于(i)字长和执行滤波器运算的算术类型，(ii)用来量化滤波器系数和变量的方法，以及(iii)滤波器的结构。根据这些因素，设计者可以评估有限字长对滤波器性能的影响，如果需要，可以采取补偿措施。某些影响可能并不显著，这依赖于滤波器是如何实现的。例如，当用高级语言在大型计算机上实现时，系数量化和舍入误差并不重要。而对于实时处理，用有限字长（典型的是8位、12位、和16位）来表示输入、输出信号、滤波器系数和算术运算结果。在这些情况下，分析量化效应对滤波器性能的影响通常是很有必要的。

分析IIR滤波器的有限字长对性能的影响比分析FIR的要困难得多，这是因为其反馈装置的原因。不过利用MATLAB（参见附录8B），对于特殊的滤波器可提供实践中的解决办法。在接下来的几节里，我们将按顺序讨论上面列出的四种误差来源。

有关有限字长对 IIR 滤波器以及其他的 DSP 系统性能的影响，我们将在第13章给出。

8.14.1 系数量化误差

回顾特性如下式规定的 IIR 滤波器：

$$H(z)=\frac{\sum_{k=0}^{N} b_k z^{-k}}{1+\sum_{k=1}^{M} a_k z^{-k}}$$

当系数被量化到有限位数时，例如8或者16位，量化的传递函数可以写成

$$[H(z)]_q=\frac{\sum_{k=0}^{N} [b_k]_q z^{-k}}{1+\sum_{k=1}^{M} [a_k]_q z^{-k}} \tag{8.52}$$

其中

$$[b_k]_q=b_k+\Delta b_k; \qquad [a_k]_q=a_k+\Delta a_k$$

Δb_k、Δa_k 分别是系数 b_k 和 a_k 的变化量

q 代表量化的系数

利用有限位数量化滤波器系数的主要影响，是其改变了 $H(z)$ 的 z 平面极点和零点的位置。这可能导致：

- 对于高阶滤波器，它具有尖锐的过渡带宽，极点靠近单位圆，可能引起不稳定或潜在的不稳定；
- 期望的频率响应的改变，如图 8.30 所示。

我们必须分析量化后的滤波器，以便确保滤波器的字长足以满足稳定性和频率响应的要求。在接下来的例子里，我们将说明系数量化效应对一个 IIR 滤波器频率响应的影响。系数量化对滤波器性能影响的详细分析将在第 13 章给出。

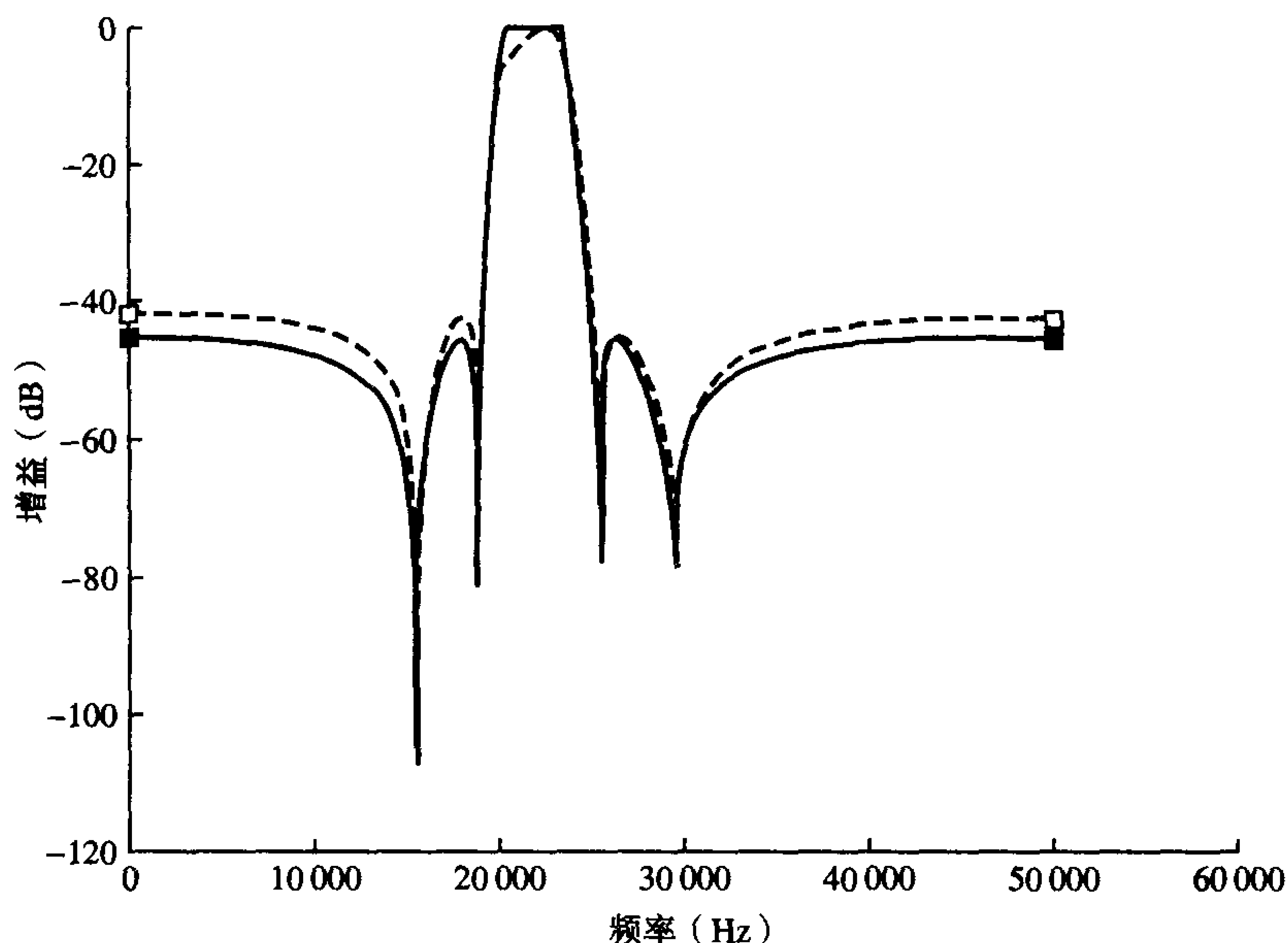


图 8.30　系数量化对频率响应的实际影响：■未量化的；□量化为 5 位

例 8.21　使用带通数字滤波器进行数字时钟恢复，波特（baud）率为 4.8 k，抽样频率是 153.6 kHz。滤波器特性由下面的传递函数来描述：

$$H(z) = \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

其中

$$a_1 = -1.957\,558 \quad 和 \quad a_2 = 0.995\,913$$

评估把系数量化到 8 位时对极点位置以及中心频率的影响。

解：

首先我们求出未量化滤波器极点的位置。每一个极点的半径 r、角度 θ 是

$$r = \sqrt{a_2}, \quad \theta = \cos^{-1}\left(-\frac{a_1}{2r}\right)$$

因此

$$r=\sqrt{0.995913}=0.99795 \quad 和 \quad \theta=\cos^{-1}\left(\frac{1.957558}{2\times 0.99795}\right)=11.25°$$

这对应的中心频率为

$$\left(\frac{11.25}{360}\right)\times 153.6\times 10^3=4.7999\text{ kHz}$$

因为有一个系数大于1，我们指定1位作为符号位，1位来表示整数，6位来表示系数的分数部分。因此，在把系数值量化成8位后，系数值变为

$$a_1=-1.957558\times 2^6=-125\ (\equiv 10000100)$$

$$a_2=0.995913\times 2^6=63\ (最大正的分数)\ (\equiv 00111111)$$

用分数形式表示，量化后的系数值是

$$a_1=-\frac{125}{64}=-1.953125;\quad a_2=\frac{63}{64}=0.984375$$

新的极点位置变为 $r=0.992156$，$\theta=10.17°$；新的中心频率变为

$$f_0=\left(\frac{10.17}{360}\right)\times 153.6\times 10^3=4.34\text{ kHz}$$

8.15 IIR 滤波器的实现

在IIR滤波器里，输出 $y(n)$ 是对每一个输入样本 $x(n)$ 来计算的。假定采用二阶直接型的串联实现结构，关键的滤波方程是

$$y(n)=\sum_{k=0}^{2} b_k x(n-k)-\sum_{k=1}^{2} a_k y(n-k)$$

这个方程很清晰的表明了如果要应用这个滤波器，我们需要如下部件：

- 存储器（例如ROM），用来储存滤波器系数；
- 存储器（例如RAM），用来存储现在和过去的输入 $\{x(n), x(n-1),\ldots\}$ 和输出 $\{y(n), y(n-1),\ldots\}$；
- 硬件或软件乘法器；
- 加法器或者算术逻辑单元。

在现代实时DSP里，DSP处理器可以有效地执行滤波运算，例如TMS320C50。这些处理器在板上带有所有基本模块，包括内置的硬件乘法器。在某些应用中，标准8位或者16位微处理器，例如摩托罗拉的6800或者68000族，它们提供了很具有吸引力的、可供选择的实现手段。除了信号处理硬件以外，设计者还必须给数字硬件提供适当的输入-输出接口（例如模拟-数字-模拟转换），这依赖于数据源和接收器的类型。这种方法可以将其描述为硬件实现。

在批处理或者离线处理时，利用合适的高级语言来实现滤波器。在这种情况下，滤波器常常是用高级语言（例如C或者FORTRAN）来实现的，并且在通用的计算机上运行，例如个人电脑或者大型计算机，其中的所有基本模块已经构造好。因此，批处理可以将其描述为一个纯软件实现。

运算量

设计者必须分析数字滤波器的运算量对所用处理器的影响。数字滤波器基本的运算是相乘、相加、累加和延时或移位。例如一个由二阶部分组成的滤波器通常将要求4次乘法、4个加法、一些移位和存储。如果滤波是实时执行的，例如以44.1 kHz（对于数字音频），则算法运算必须

每 1/(44.1 kHz)执行一次。对于其他的管理操作，例如取出输入数据，存储或输出滤波后的样本，像其他程序操作一样，也必须规定出允许量。

8.16　IIR 数字滤波器详细的设计举例

这个例子将用来解释本章的一些概念。特别是，我们将看到设计的五个步骤是如何应用的。

步骤 1：滤波器规范

利用软件包和基于TMS320C50的目标板来设计和实现低通数字滤波器，要求满足下面的规范：

抽样频率	15 kHz
通带	0 ~ 3 kHz
过渡带宽	450 Hz
通带波纹	0.5 dB
阻带衰减	45 dB

注意：目标板有一个 12 位的 ADC 和 12 位的 DAC。

步骤 2：系数计算

利用IIR 滤波器的软件设计程序（在指导手册的CD上），我们将发现，通过双线性变换法得到的四阶椭圆滤波器，满足上面的性能规范。设计程序的输出表总结如下：

	分母 A_k	分子 B_k
1	1.000000E+00	5.846399E-02
2	-1.325263E+00	1.359507E-01
3	1.480202E+00	1.820297E-01
4	-7.841098E-01	1.359506E-01
5	2.339270E-01	5.846398E-02

极点		系数	
实部	虚部	z^{-1}	z^{-2}
0.247967	0.836885	-0.495935	0.761864
0.414664	0.367559	-0.829328	0.307046

零点		系数	
实部	虚部	z^{-1}	z^{-2}
-0.337859	0.941197	0.675718	1.000000
-0.824828	0.565383	1.649656	1.000000

根据上面的表，给出直接形式的滤波器传递函数为

$$H(z) = \frac{0.05846399 + 0.1359507z^{-1} + 0.1820979z^{-2} + 0.1359506z^{-3} + 0.05846398z^{-4}}{1 - 1.325263z^{-1} + 1.480202z^{-2} - 0.7841098z^{-3} + 0.233927z^{-4}}$$

步骤 3：实现结构

如同前面解释过的那样，$H(z)$的直接实现形式对有限字长的许多不利影响（例如系数量化误差）特别敏感，所以把$H(z)$分解成更小的部分，然后把它们用串联或者并联的形式连接起来，这是非常重要的。假设在串联结构里，$H(z)$被分解成两个二阶部分 $H_1(z)$和 $H_2(z)$：

$$H(z) = H_1(z)H_2(z)$$

其中

$$H_1(z) = \frac{b_{01} + b_{11}z^{-1} + b_{12}z^{-2}}{1 + a_{11}z^{-1} + a_{21}z^{-2}}$$

$$H_2(z) = \frac{b_{02} + b_{12}z^{-1} + b_{22}z^{-2}}{1 + a_{12}z^{-1} + a_{22}z^{-2}}$$

在图8.31里描绘了实现图，其中每一个滤波器部分都是用一个标准的二次结构实现的。对应的定义滤波器的差分方程组如下：

滤波器部分 1

$$w_1(n) = (1/s_1)x(n) - a_{11}w_1(n-1) - a_{21}w_1(n-2)$$

$$y_1(n) = b_{01}w_1(n)s_1/s_2 + b_{11}w_1(n-1)s_1/s_2 + b_{21}w_1(n-2)s_1/s_2$$

滤波器部分 2

$$w_2(n) = y_1(n) - a_{12}w_2(n-1) - a_{22}w_2(n-2)$$

$$y_2(n) = b_{02}w_2(n)s_2 + b_{12}w_2(n-1)s_2 + b_{22}w_2(n-2)s_2$$

系数 a_{ij} 和 b_{ij} 的精确值依赖于我们对 $H(z)$ 的多项式的分子和分母如何组对，以及实现多项式的二阶滤波器部分是如何排序的。最好的组对和排序只能通过有限字长分析来确定。

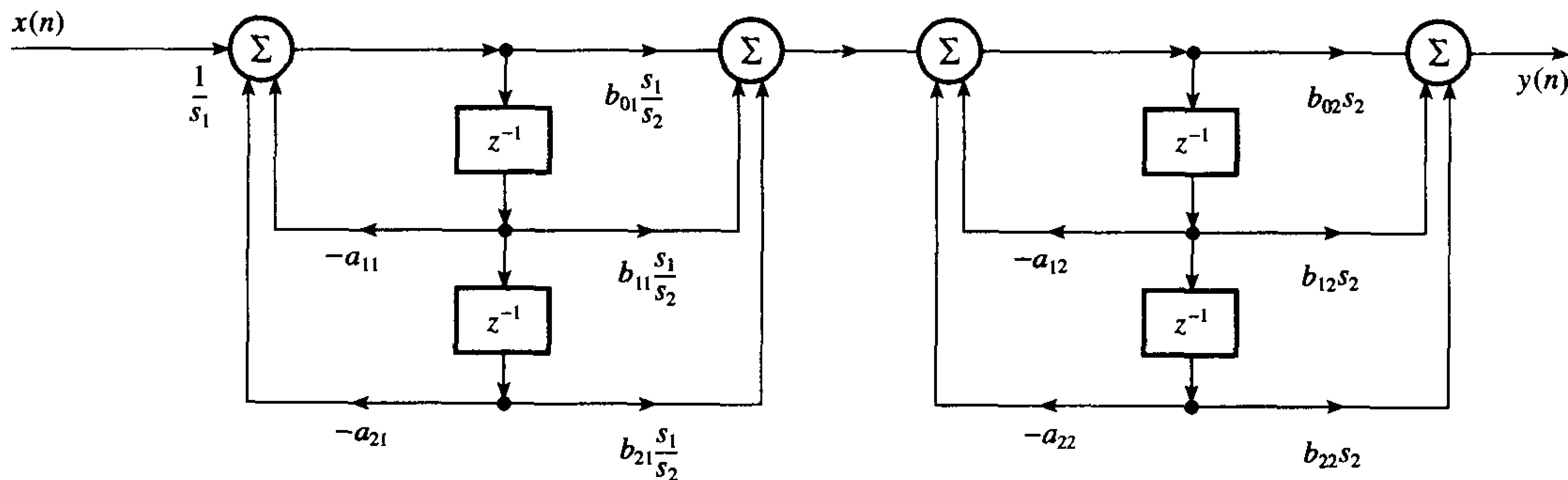


图 8.31 详细设计实例的实现结构图

步骤 4：有限字长效应分析

对于这个问题，根据给定的性能规范，我们将假设使用的算法是定点的2的补码算术，每一个系数被舍入量化到16位字长。

这里我们的主要目的是评估不同量化误差对滤波器性能的影响，以及根据信噪比确定最好的滤波器结构用于实现。考虑的误差源是（详情见第13章）

- 溢出误差
- 舍入误差
- 系数量化误差

为了避免图8.31所示的加法器输出的溢出，在如图中所示的加法器的前面，要引入一个适当的比例因子。

因为 $H(z)$ 是四阶的，它用两个二阶部分来实现，它的分子和分母因式可以用下面4种可能的方式进行组对和排序：

$$H_A(z) = \frac{N_1(z)}{D_1(z)} \frac{N_2(z)}{D_2(z)}$$

$$H_B(z) = \frac{N_2(z)}{D_2(z)} \frac{N_1(z)}{D_1(z)}$$

$$H_C(z) = \frac{N_1(z)}{D_2(z)} \frac{N_2(z)}{D_1(z)}$$

$$H_D(z) = \frac{N_2(z)}{D_1(z)} \frac{N_1(z)}{D_2(z)}$$

其中

$$N_1(z) = 1 + 0.675\,718z^{-1} + z^{-2}$$

$$N_2(z) = 1 + 1.649\,656z^{-1} + z^{-2}$$

$$D_1(z) = 1 - 0.495\,935z^{-1} + 0.761\,864z^{-2}$$

$$D_2(z) = 1 - 0.829\,328z^{-1} + 0.307\,046z^{-2}$$

四个可能的滤波器结构的每一个都有不同的比例因子，以及不同的信号舍入误差。这个步骤的目标是根据信噪比性能方面来确定最好的组对和排序。溢出和舍入误差是紧密联系的，所以伸缩变换和舍入分析应该同时进行的。

利用有限字长分析程序，可以得到对于上面四个可能的滤波器的比例因子，它们是基于L_1、L_2和L_∞的范数，在表 8.1 里进行了总结。在这个例子里，我们利用了L_1的范数。对于一个以两个标准二阶部分串联实现的四阶滤波器，在伸缩变换以后，输出端的舍入噪声为

$$\sigma_o^2 = \frac{q^2}{12}[3s_1^2\|H_1(z)H_2(z)\|_2^2 + 5s_2^2\|H(z)\|_2^2 + 3]$$

其中q是量化步长或舍入，$\|\cdot\|_2^2$是L_2范数的平方。$H_1(z)$是第一级滤波器的传递函数，$H_2(z)$是第二级滤波器的传递函数，s_1是第一个滤波器阶段的伸缩的比例因子，s_2是第二级滤波器阶段的比例因子。

表 8.1　四个滤波器构造的比例因子

滤波器	比例因子	L_1	L_2	L_∞
A	s_1	5.524 844	1.608 890	4.379 544
	s_2	11.821 571	3.677 381	7.262 393
B	s_1	2.479 158	1.359 467	2.175 539
	s_2	18.908 47	10.880 490	12.548 114
C	s_1	2.479 158	1.359 467	2.175 539
	s_2	11.821 571	10.880 490	7.262 393
D	s_1	5.524 844	1.608 890	4.379 544
	s_2	18.908 47	5.727 459	12.548 114

当每一个系数量化成 16 位（伸缩后）时，四个可能滤波器构造中的每一个的噪声性能如表 8.2 所示。很显然滤波器 B 具有最好的舍入噪声性能。伸缩后的传递函数由下式给出：

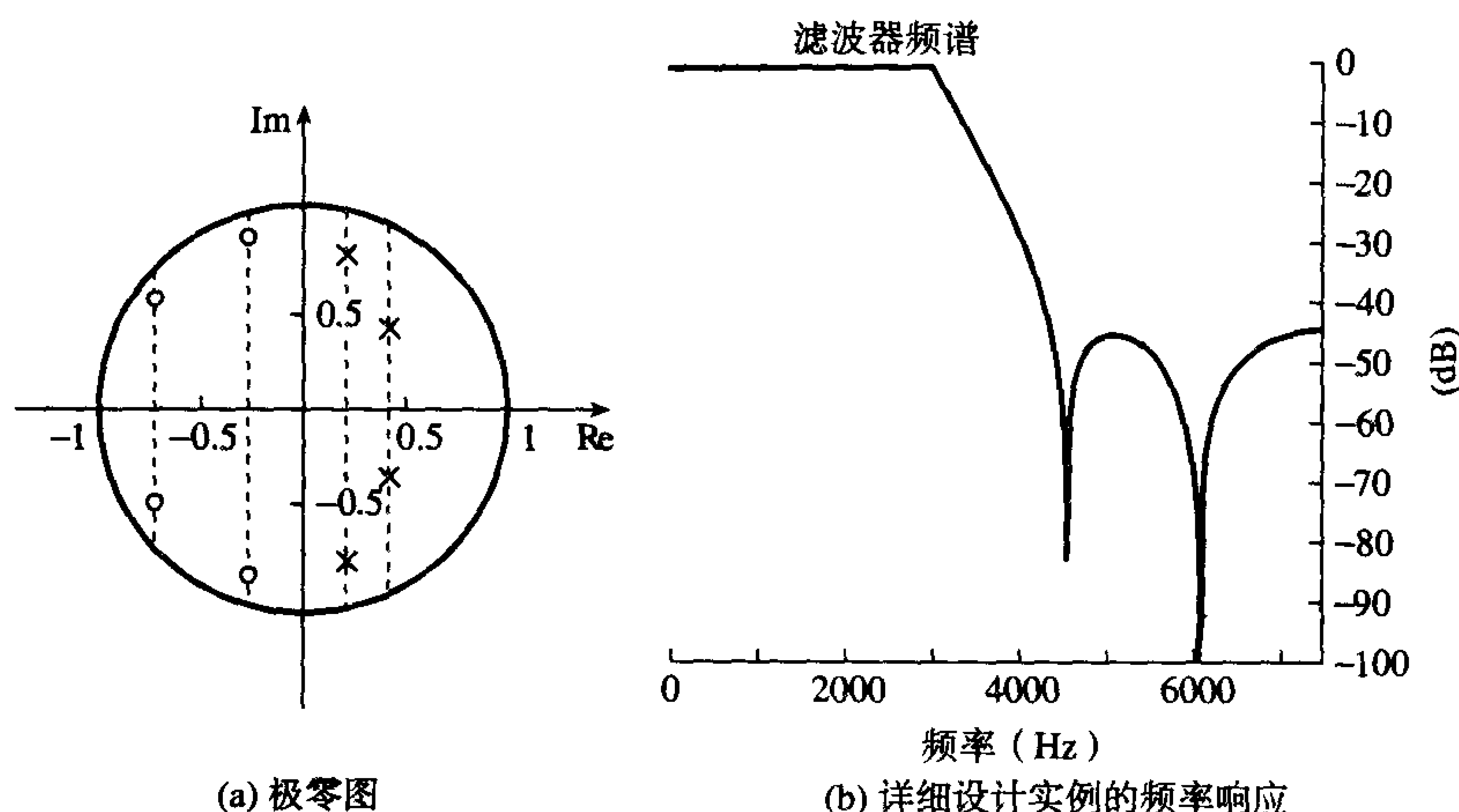
$$H(z) = H'_B(z) = \frac{s_1}{s_2} \frac{N_2(z)}{D_2(z)} \frac{N_1(z)}{D_1(z)} s_2$$

$$= 0.131\,113\,6 \frac{1 + 1.649\,656z^{-1} + z^{-2}}{1 - 0.829\,328z^{-1} + 0.307\,046z^{-2}} \times \frac{1 + 0.675\,718z^{-1} + z^{-2}}{1 - 0.495\,935z^{-1} + 0.761\,864z^{-2}} 10.880\,490$$

表 8.2　四个滤波器结构的舍入噪声性能的比较

滤波器	噪声功率
A	$703q^2$
B	$326.378q^2$
C	$382.32q^2$
D	$570.453q^2$

接下来，我们分析系数量化误差的效应。特别是，我们检查那些给出的系数字长是否满足稳定和频率响应的规范。当极点不是非常靠近单位圆时，16位的系数字长对保持稳定是足够的。例如，对于第一个滤波器部分，FWA程序已经表明3位就足以满足稳定的要求，把系数量化成16位，仅改变极点半径约0.000 48%。FWA程序也表明，仅要求12位就可以保证频率响应在容差限度内。采用16位的系数字长时，滤波器的响应实际上和未量化的滤波器的响应是一样的。图8.32描绘了未量化的滤波器的频率响应和极零图。



(a) 极零图　　(b) 详细设计实例的频率响应

图 8.32　未量化的滤波器的频率响应和极零图

利用乘积之和的双长度累加以及累加后的量化使舍入噪声的影响达到最小。

步骤5

量化后的系数（通过给伸缩后的系数乘以2^{15}得到）被输入到附录中列出的TMS320C25 IIR程序中。第12章给出了一个更详细的IIR滤波程序以及相应的讨论。

8.17　小结

IIR滤波器的设计可以分成五个互相关联的步骤（参见图8.1）。滤波器规范通常依赖于应用，但通常应该包括带沿、幅度响应的容差范围、抽样速率和I/O要求等细节。对于具有标准特性的滤波器，通过BZT能有效得到所要求的满足幅度响应规定的系数。这个方法和其他有用的系数计算法一样，在下面的部分中将进行描述，并且有许多相关的例子。为了保证极点和零点位置的改变而使有限字长效应较小，高阶IIR滤波器用二阶和一阶部分串联或并联连接来实现。对每一部分的输入用伸缩（比例）变换来防止内接点中的溢出，这是非常必要的。

IIR数字滤波器的性能受到实现中所用的位数数目的限定。四个通常的误差是由以下方面引起的：(1)输入量化，(2)系数量化，(3)乘积舍入，(4)加法溢出。分析它们对滤波器性能的影响以及消除（如果可能）或使其最小化的技术已经给出。系数量化必须满足使系数量化对频率响应的影响达到最小，同时还要防止滤波器可能产生的不稳定性。IIR滤波器的稳定性总是要关心的。一个在无限

精度中应用的滤波器是稳定的，然而当应用在有限精度中时可能会变成不稳定的。例如在高保真音频工作中，处理低频音频信号24位的系数是必需的。在许多其他情况下，用16位或者更多些位来表示系数，并用双长度累加器来执行算术运算足以使有限字长效应达到最小。

由有限精度算术运算而引起的截尾或舍入误差，在滤波器中产生了非线性效应，例如极限环，甚至滤波器在没有输入（或者一个常数输入时）时也会产生振荡。舍入误差对滤波器性能的影响可以根据滤波器输出的 SNR 来定量化。由于舍入误差而引起的 SNR 的减少，可利用误差频谱整形（ESS）方法来弥补（参见第13章）。这些方法的基本效果是使滤波器极点对舍入误差的放大效应失效。为此付出的代价是乘法和加法数目的增多，尽管具有整数系数的一阶 ESS 在计算上是有效的。

在指导手册的CD里提供了设计程序，可以使设计者计算滤波器系数和分析有限字长对滤波器性能的影响（详情请参见前言）。

8.18　在数字音频和装置里的应用例子

本节里给出了一些应用的综述，其中将用到 IIR 滤波器或者它适合使用。

8.18.1　数字音频

在数字音频中的许多领域，特别是在具有高质量数字源的系统中，例如CD播放器和DAT，数字滤波器有许多用处。此时和其他尽可能多的信号处理的操作一样，数字化实现也是很有意义的。DSP也能使产生例如音乐厅、爵士乐厅、迪斯科等地方的声音特性变为可能。在一个利用了IIR滤波器的数字音频的应用中，包括图形均衡、音调控制、通道均衡、ADC/DAC里的噪声整形和频带分离。

例如在数字图形均衡器里，IIR 滤波器用来把整个声音频率范围分离成不同频段，使得再生声音的广泛音调可以调节到合适的程序，而不必通过低音和高音的控制。一个典型的五波段图形均衡器把整个音频范围分成五个频段，中心频率是 100 Hz、330 Hz、3.3 kHz、10 kHz 和 16 kHz，在每一个频段内允许可调整的信号电平范围是 ±10 dB。

在图 8.33 里给出了图形均衡的简单的滤波排列。

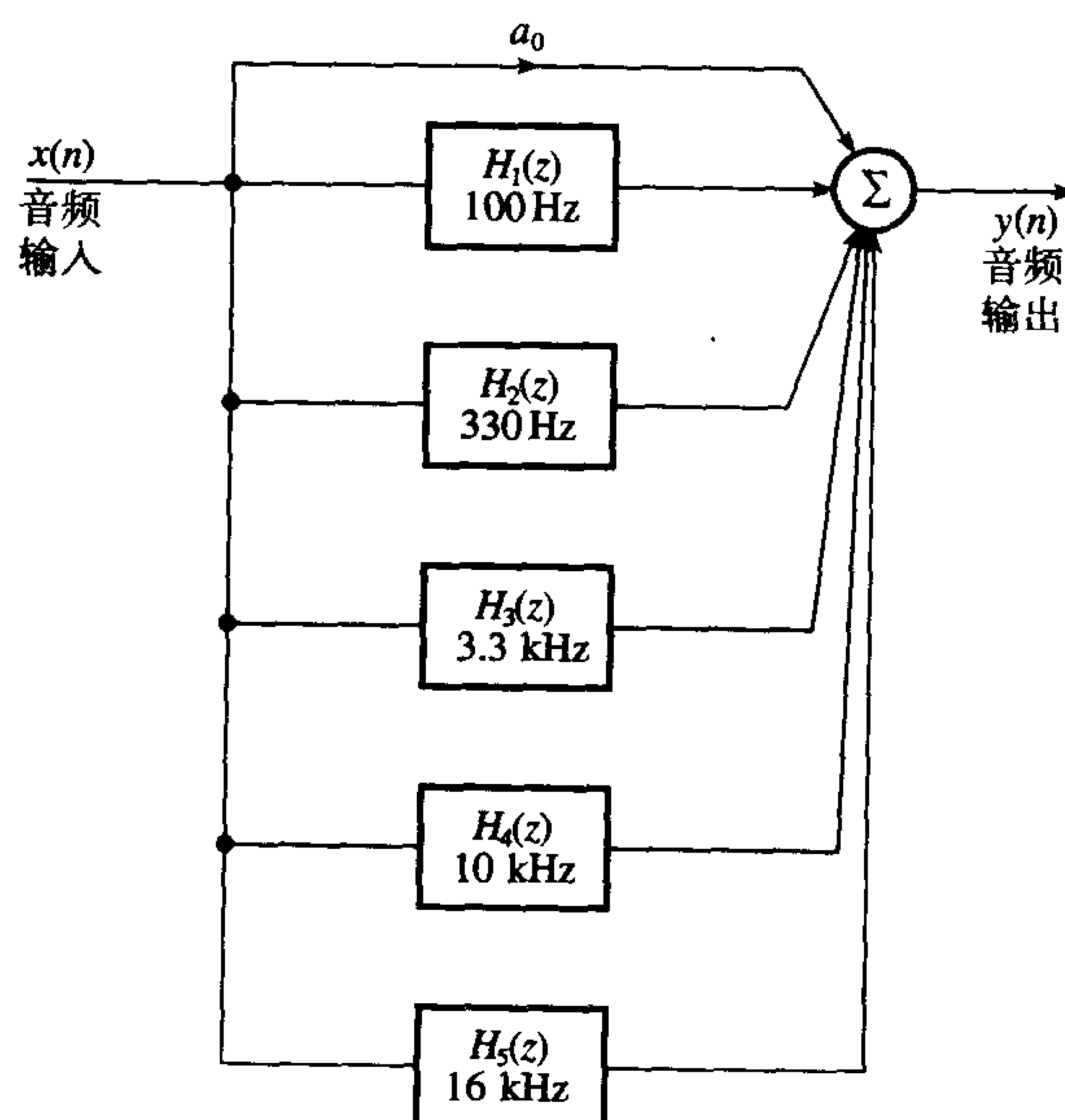


图 8.33　一个全数字图形器的简化框图。它的主要部件是一组并行的具有不同中心频率的IIR滤波器。每一个滤波器的增益是分别可调节的，例如用一个滑动的电位器，调节范围为±10 dB

8.18.2 数字控制

随着对DSP好处的认识的加深以及造价低廉的处理器的出现，控制器正在被数字化地实现，以获得更好的精度和适应性。图8.34显示了一个模拟的设备（可能例如是一个汽车或者摩托）$H(s)$的数字控制的原则。一般来说，数字控制器具有 IIR 的特性。

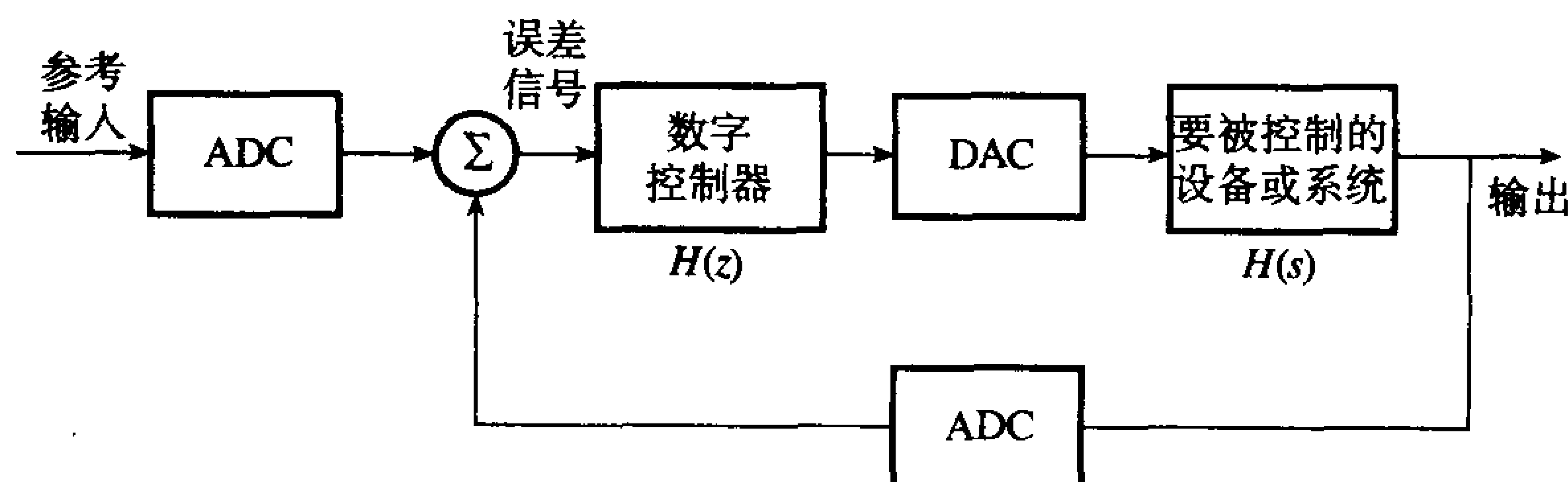


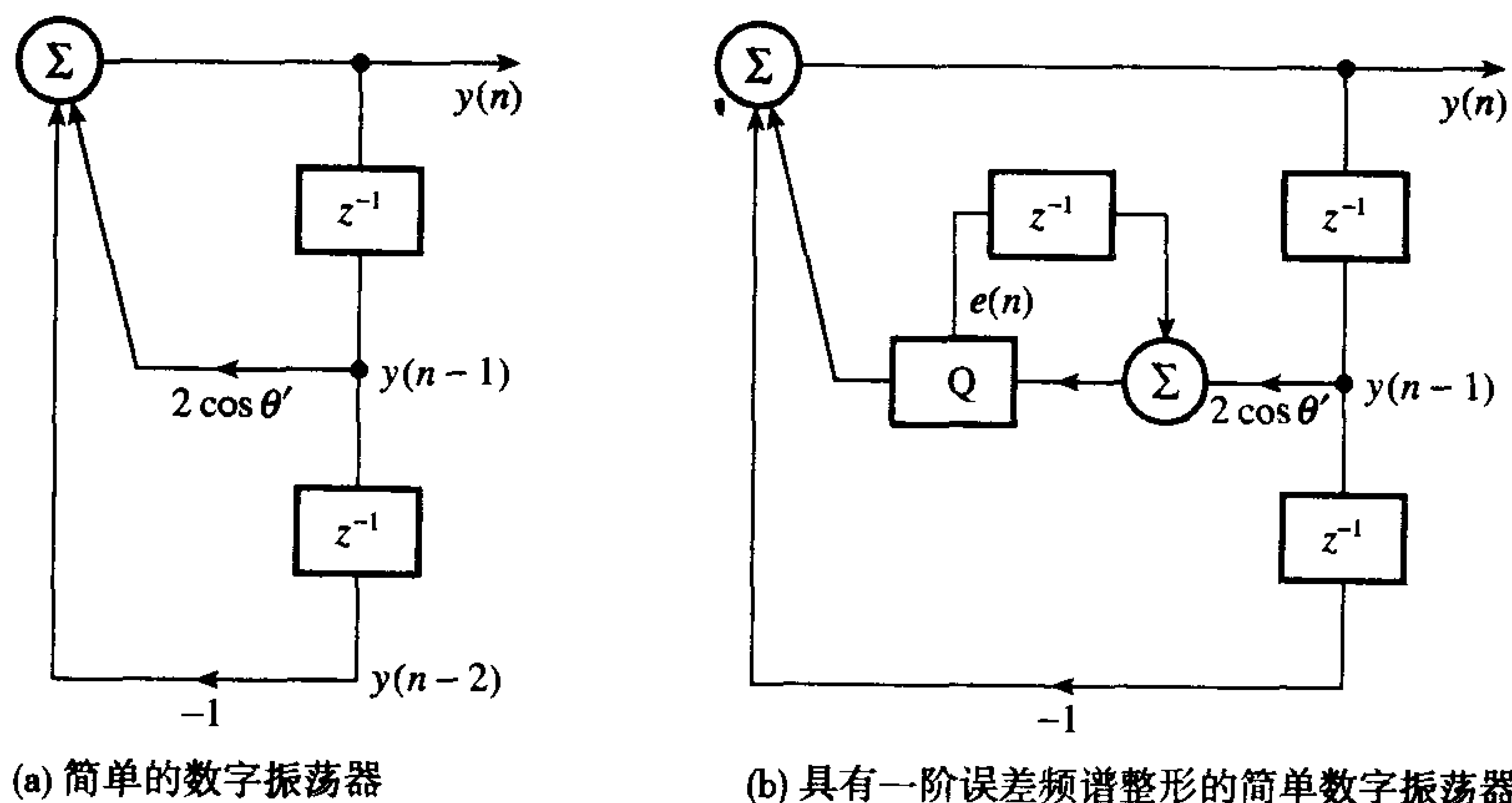
图 8.34 一个模拟设备的数字控制原理

8.18.3 数字频率振荡器

IIR 滤波器用来产生精确的波形以代替传统的查表方法。这个方法利用了这样的事实：一个具有靠近单位圆的极点的IIR 滤波器本质上是不稳定的。在图8.35(a)中给出了一个简单的正弦波形振荡器。这个 IIR 滤波器的极点位于 $e^{\pm j\theta}$，振荡器的频率如下式给定：

$$\theta' = w_0 T_B$$

其中 T 是抽样周期。滤波器系数 $2\cos\theta'$ 是通过取 $2^B \times 2\cos\theta'$（B 是位数）的整数部分而约成一个整数的。



(a) 简单的数字振荡器 (b) 具有一阶误差频谱整形的简单数字振荡器

图 8.35 简单的数字振荡器

在使用IIR 滤波器的数字波形发生器中，主要的问题是涉及到有限字长效应。例如，系数量化将导致频率分布不均匀，而乘积量化会导致舍入误差的累加，这可能致使波形发生器失效。不过，采用ESS 技术可以使这些误差最小。图 8.35(b)给出了一个利用 ESS 技术（Abu-el-Haija and Al-Ibrahim, 1986）的振荡器，该技术可以使舍入噪声影响显著降低。

8.19 在电信中的应用举例

IIR 滤波器因为它的特性而被广泛应用于数字通信领域。在数字电话中（Feeney et al., 1971），PCM 允许许多声音通道同时传输。每一个通道在限带以后以 8 kHz 抽样，且利用 A 律或者 μ 律编码。在接收端，PCM 数据被转换回模拟信号，且进行抗镜频滤波。数字 IIR 滤波器可以在发射和接

收端提供必要的滤波（参见图 8.36）。在这种情况下，滤波以较高的抽样率进行，例如 32 kHz，接着从线性码转化成标准 PCM 码。

在下两个小节中，我们将讨论数字通信领域中 IIR 滤波器的两个特殊应用，一个用于按钮式数字电话，另一个用于数据通信中的时钟恢复。

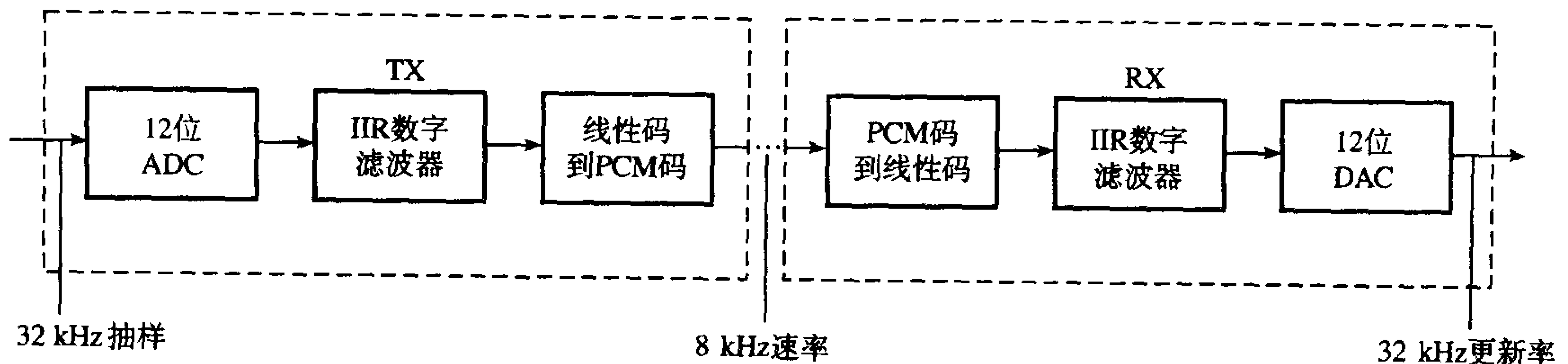


图 8.36　PCM 通道表明了 IIR 滤波器对主抗混叠滤波器（TX 端）和抗像频滤波（RX 端）的可能应用

8.19.1　数字电话的按钮式产生和接收

IIR滤波器一个好的应用是全数字双音调多频率按钮式接收器（Jackson et al., 1968; Mock, 1985）。

在现代电话系统里，要求建立通信的信息，并且为了便于维护和指示，这些信息通常由一个多频码提供。典型的如电话机产生两个音调，一个是低频音调，另一个是高频音调（参见图 8.37）。

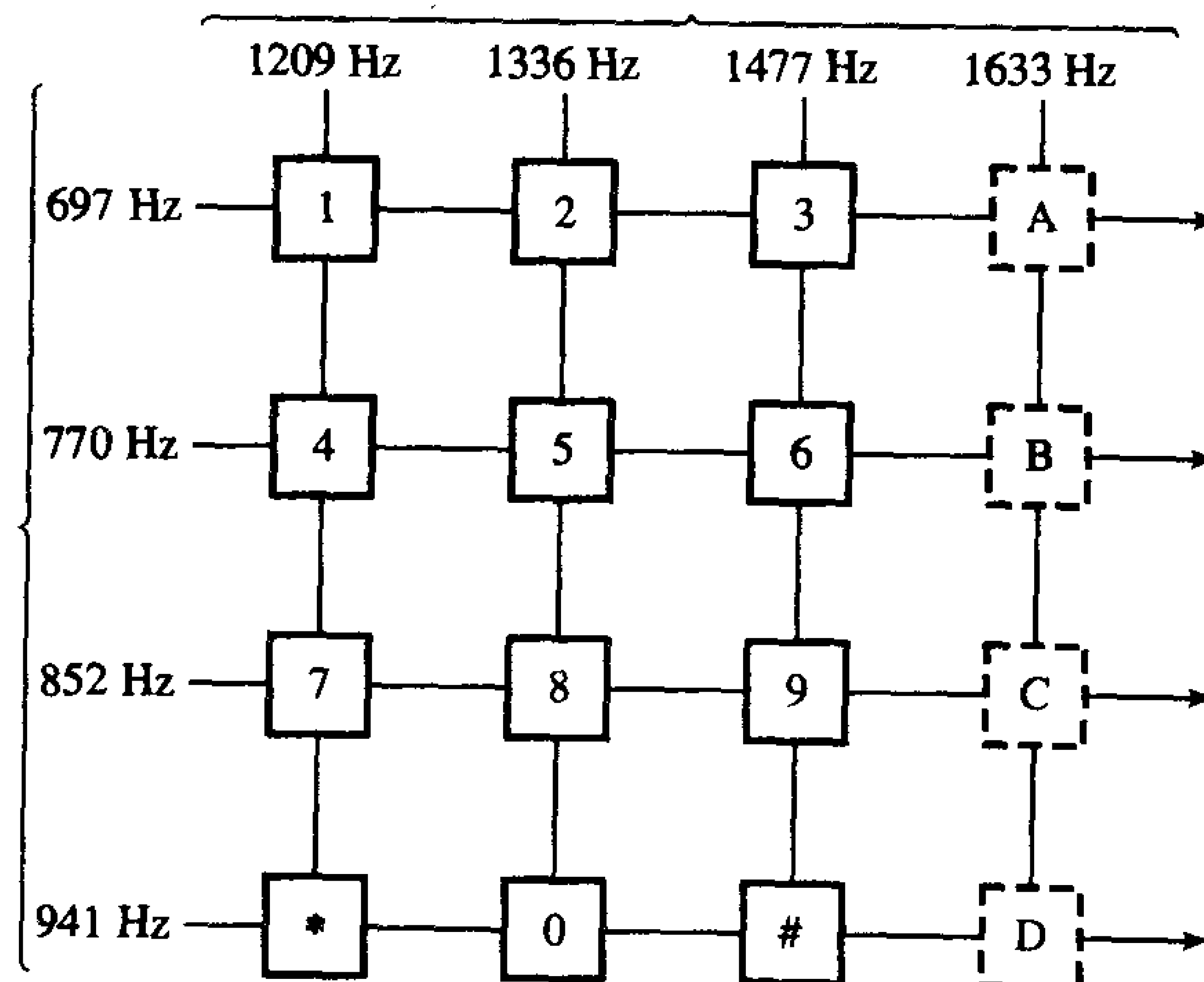


图 8.37　一个 4 × 4 键的按钮式电话的简化图。虚线画的按钮是不可用的。按一个按钮产生一对音调，一个来自低频组，另一个来自高频组。例如，按 9 产生 852 Hz 和 1477 Hz 的音调（after Mock, 1985）

音调产生器可以利用一对可编程的二阶 IIR 振荡器实现（参见图 8.38）。当按下一个按钮时，拨号数字的编码用来从 ROM 中选择合适的滤波器系数，并且初始化条件，产生一对音调（一个是高频音调，一个是低频音调）。音调相加产生按钮式信号。利用数字正弦波形产生器，按钮式音调产生器的性能通过误差反馈方法而得到了改善。

在接收端，信息以 8 kHz 的速率被数字化，接着被前端带通滤波器分隔成一个低频和高频通带。为了检测一个音调是否出现，我们进行电平检测。它是通过把带通滤波和全波形整流结合在一起后接一个低通滤波器来实现的。为了检测是低频中的哪一个音调出现，由四个 BPF 中的两组将低频带分离成四个频带；高频带也是如此。最后得到的 8 个电平将传输到判决逻辑单元来确定接收到的编码。

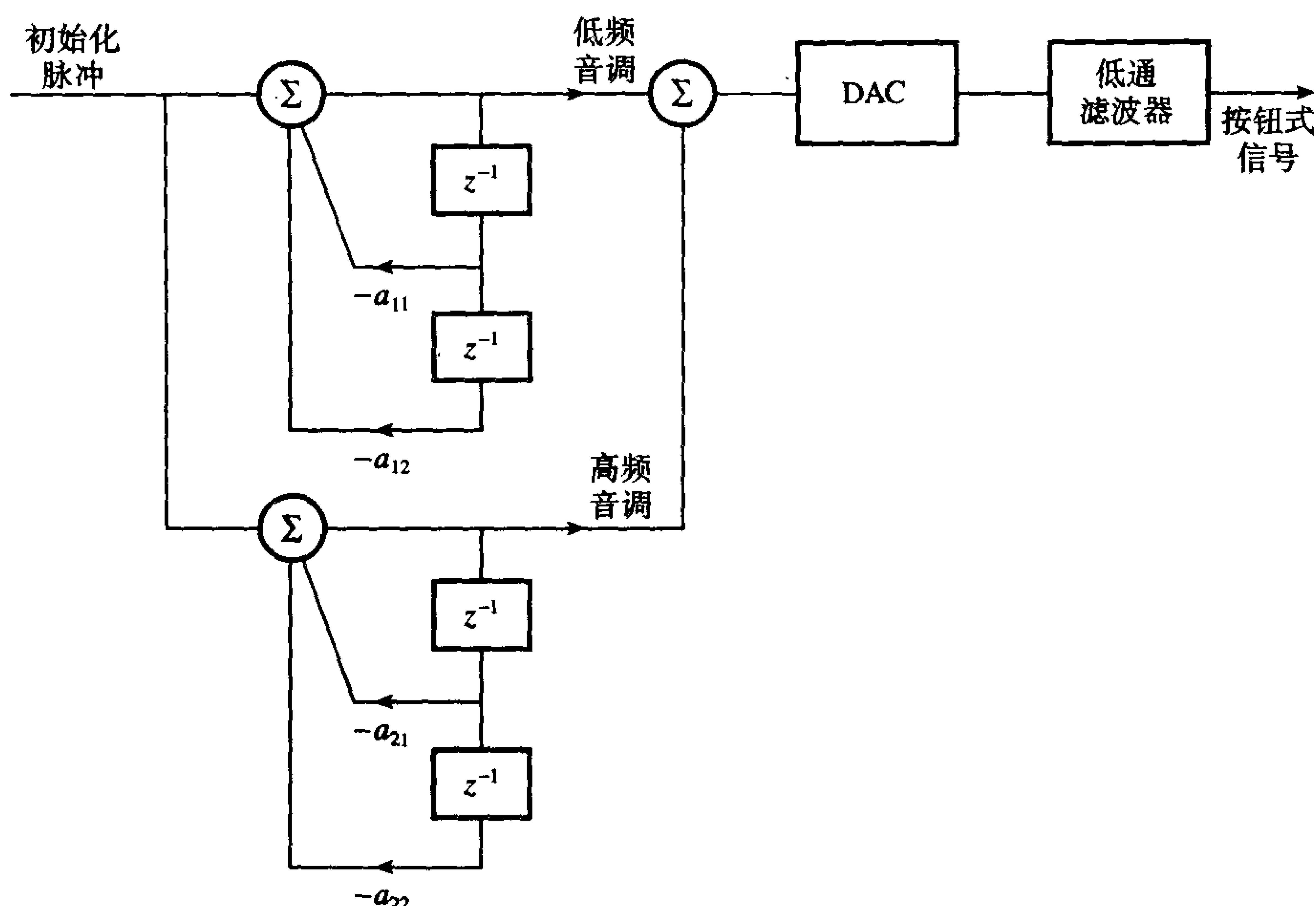


图 8.38 按钮式音调产生器（after Mock, 1985）。拨号数字的编码用来选择滤波器系数并且初始化条件，还将产生振荡器的频率

8.19.2 数字电话：利用 Goertzel 算法的双音多频（DTMF）检测

Goertzel算法可以用做标准IIR滤波器的替代品，以检测DTMF音调（Mock, 1985; Marven, 1990; Chen, 1996; Texas Instruments, 1997）。Goertzel 算法是一种特殊的 IIR 滤波器的离散傅里叶变换（DFT）的实现。图 8.39 描绘了基于 Goertzel 算法的 DTMF 检测方法的框图。它由一排平行的 8 对 Goertzel 滤波器组成。每一个滤波器对检测一个 DTMF 音调和它的二次谐波。要求利用二次谐波来区分语音和 DTMF 音调。语音有显著的偶数阶谐波，而 DTMF 信号则没有。对每个滤波器的输出进行平方，以得到 8 个 DTMF 频率处以及二次谐波的信号强度的度量。从高频和低频组得到的最强的信号用来确定接收的数字。

每一个 Goertzel 滤波器都是高 Q、窄带、二阶、带通 IIR 滤波器，特性由下面的传递函数给定（参见图 8.40）：

$$H_k(z) = \frac{1 - W_N^k z^{-1}}{1 - 2\cos\left(\frac{2\pi k}{N}\right)z^{-1} + z^{-2}} \tag{8.53a}$$

其中

$$W_N^k = \exp\left(-\frac{2\pi kj}{N}\right)$$

滤波器的差分方程为

$$v_k(n) = 2\cos\left(\frac{2\pi k}{N}\right)v_k(n-1) - v_k(n-2) + x(n),\ n = 0, 1, \ldots, N \tag{8.53b}$$

$$y_k(n) = v_k(n) - W_N^k v_k(n-1) \tag{8.53c}$$

其中

$$w_k(-1) = w_k(-2) = 0$$

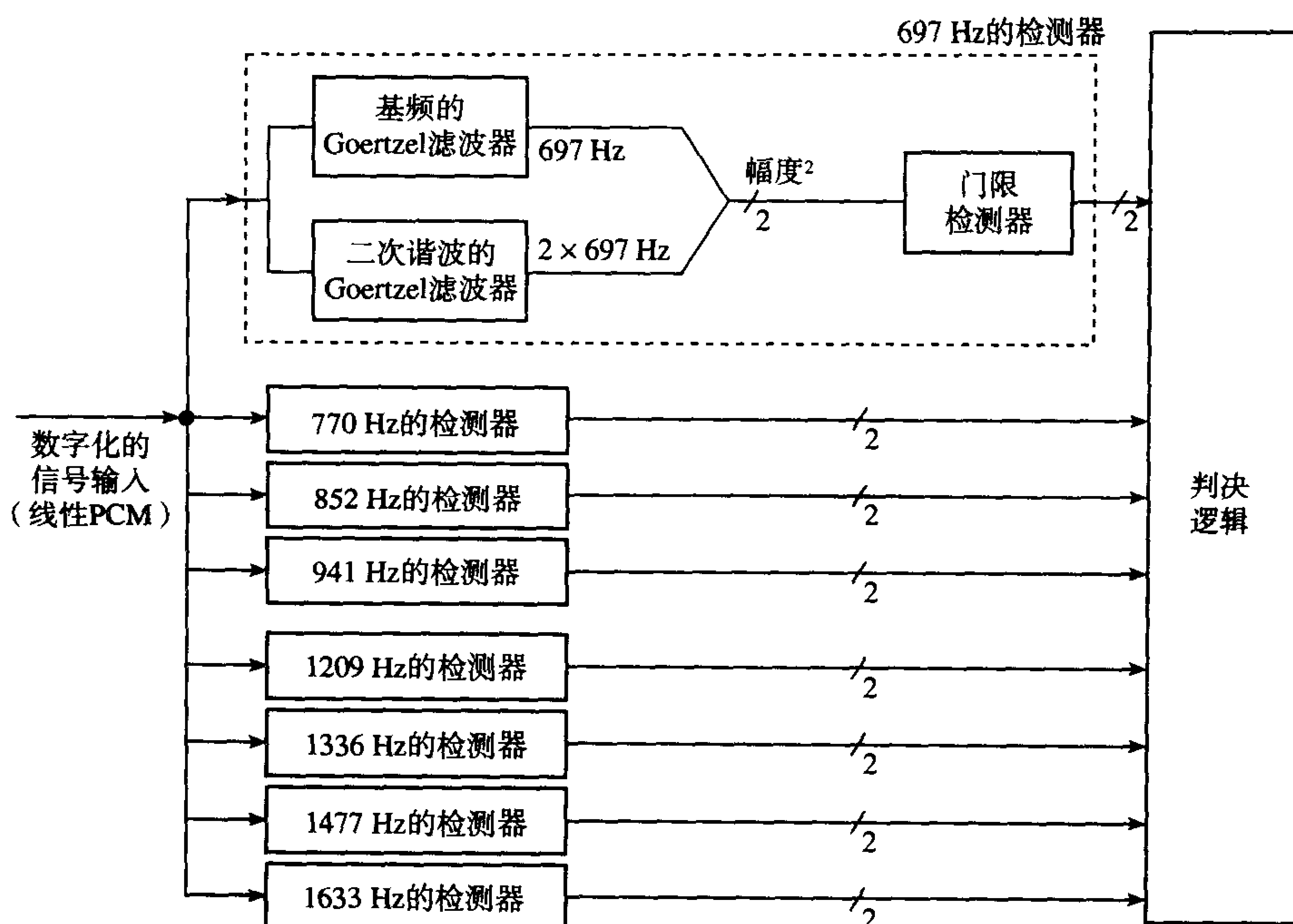


图 8.39　利用 Goertzel 滤波器的 DTMF 译码的原理（after Mock, 1985）

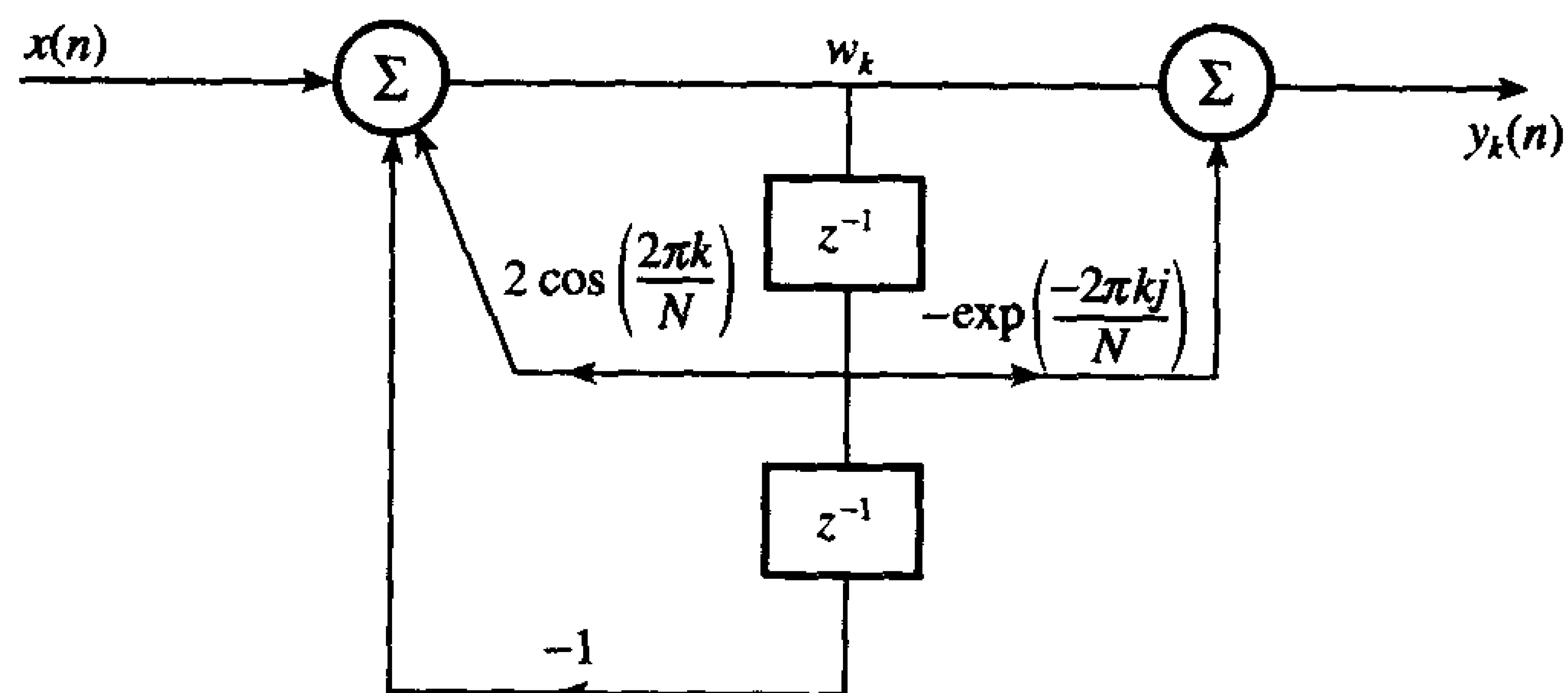


图 8.40　二阶 Goertzel 滤波器的结构

对于DTMF音调检测，只要求DTMF信号的幅度（相位信息忽略），所以GA被修正成仅产生幅度平方输出：

$$|y_k(N)|^2 = v_k^2(N) + v_k^2(N-1) - 2\cos\left(\frac{2\pi k}{N}\right)v_k(N)v_k(N-1) \tag{8.53d}$$

修正的DTMF算法（8.53b式和8.53d式）只要求一个实系数，即

$$2\cos\left(\frac{2\pi k}{N}\right)$$

利用这个系数来计算每个DTMF信号的幅度，避免了复数运算。滤波器的输出8.53d式对每个DTMF音调当 $n = N$ 时只计算一次，也就是在滤波器反馈通路的迭代的最后只计算一次。

Goertzel算法的优势包括对每一个DTMF频率仅需要一个实系数就可以确定信号的幅度，占用的存储空间少，执行速度非常快。不像FFT，它不需要在处理前等待全部数据集，而是每来一个样本它就处理一次。利用FFT，N 的值为了效率起见通常受限制（一般是2的幂）。而在Goertzel算法中，N 可以采用任何整数值，尽管 N 的选择是在频率分辨力和计算时间之间进行折中。

DFT的长度N和频率单元数目k，决定了滤波器系数的值以及检测的频率。

DFT的长度N、离散频率存储单元k、抽样间隔T以及DTMF音调频率f_k有如下关系：

$$f_k = \frac{k}{NT}$$

抽样频率和音调频率是依照国际标准设置的。DFT的长度N可以变化。表8.3列出了可能解码方法的参数（Mock, 1985）。

表8.3 DTMF解码方法的参数（after Mock,1985）

抽样率： 8 kHz

DFT长度： 205（一次谐波），210（二次谐波）

DTMF频率（Hz）	离散频率单元 k（一次谐波）	离散频率单元 k（二次谐波）
697	18	35
770	20	39
852	22	43
941	24	47
1209	31	61
1336	34	67
1477	38	74
1633	42	82

应该指出的是，Goertzel滤波器具有靠近单位圆的极点，它对有限字长效应是非常敏感，这些效应不应该被忽略。另外，如果要求解的频率点的数目相当大，那么FFT可能更合适。

例8.22

(a) 一个数据序列$x(n)$（$n = 0, 1,..., N-1$）的离散傅里叶变换（DFT）定义为

$$X(k) = \sum_{m=0}^{N-1} x(m)W_N^{km}, \quad k = 0, 1, \ldots, N-1$$

其中W_N^{km}是旋转因子。

(i) 从上面的方程出发，证明用于DTMF音调检测的Goertzel滤波器的z平面传递函数$H_k(z)$可以用下列递归形式表示为

$$H_k(z) = \frac{1 - W_N^k z^{-1}}{1 - 2\cos\left(\frac{2\pi k}{N}\right)z^{-1} + z^{-2}}$$

(ii) 推导Goertzel滤波器的幅度平方输出$|y_k(n)|^2$在离散时间$n = N$时的表达式，证明在修正的Goertzel算法中不要求复数运算。

(b) 按键电话系统的DTMF音调检测方法是基于总结在表8.3中的性能规范，而且它利用了一个二阶Goertzel滤波器。

如果拨打的数字是“99”，那么在接收端计算Goertzel滤波器的系数，以解码出数字。

解：

(a) (i) 现在

$$X(k) = \sum_{m=0}^{N-1} x(m)W_N^{km}$$

利用旋转因子的周期性，我们可以写出如下所示的 DFT 方程（因为$W_N^{-kN}=1$）：

$$\begin{aligned} X(k) &= W_N^{-kN}\sum_{m=0}^{N-1}x(m)W_N^{km} \\ &= \sum_{m=0}^{N-1}x(m)W_N^{-k(N-m)} \end{aligned} \tag{8.54}$$

8.54 式具有和卷积方程同样的形式。因此，如果我们定义数据系列 $y_k(n)$，可以得出

$$y_k(n) = \sum_{m=0}^{N-1}x(m)W_N^{-k(n-m)} \tag{8.55}$$

可以将 $y_k(n)$看成 FIR 滤波器的输出，这个滤波器的输入数据系列为 $x(m)$，N 个系数 $h_k(n)$由下式给出：

$$h_k(n) = W_N^{-kn} \tag{8.56}$$

比较 8.54 式和 8.55 式，可以看出当 $n=N$ 时，滤波器输出频率单元 k 给出的 DFT $X(k)$为

$$X(k) = y_k(n)\big|_{n=N}$$

滤波器的 z 平面传递函数为

$$\begin{aligned} H_k(z) &= \sum_{n=0}^{\infty}W_N^{-kn}z^{-n} \\ &= \frac{1}{1-W_N^{-k}z^{-1}} \end{aligned} \tag{8.57}$$

这是一个一阶滤波器，它具有一个在单位圆$z=W_N^{-k}$上的单复极点（由于旋转因子是复数）。为了避免复数运算，单极点被组合成复共轭极点对来生成一个二阶滤波器部分。这可以通过对 8.57 式乘以

$$\left(\frac{1-W_N^kz^{-1}}{1-W_N^kz^{-1}}\right)$$

得到

$$H_k(z) = \frac{1-W_N^kz^{-1}}{1-2\cos\left(\dfrac{2\pi k}{N}\right)z^{-1}+z^{-2}}$$

(ii) 从图 8.40 可以得到 Goertzel 滤波器的两步差分方程：

$$v_k(n) = 2\cos\left(\frac{2\pi k}{N}\right)v_k(n-1)-v_k(n-2)+x(n)$$

$$y_k(n) = v_k(n)-W_N^kv_k(n-1)$$

在 $n=N$ 时，

$$\begin{aligned} y_k(N) &= v_k(N)-W_N^kv_k(N-1) \\ &= v_k(N)-v_k(N-1)\left[\cos\left(\frac{2\pi k}{N}\right)-\mathrm{j}\sin\left(\frac{2\pi k}{N}\right)\right] \\ &= v_k(N)-v_k(N-1)\cos\left(\frac{2\pi k}{N}\right)+\mathrm{j}v_k(N-1)\sin\left(\frac{2\pi k}{N}\right) \end{aligned}$$

$$
\begin{aligned}
|y_k(N)|^2 &= (\text{实部})^2 + (\text{虚部})^2 \\
&= \left[v_k(N) - v_k(N-1)\cos\left(\frac{2\pi k}{N}\right)\right]^2 + \left[v_k(N-1)\sin\left(\frac{2\pi k}{N}\right)\right]^2 \\
&= v_k^2(N) - 2v_k(N)v_k(N-1)\cos\left(\frac{2\pi k}{N}\right) \\
&\quad + v_k^2(N-1)\cos^2\left(\frac{2\pi k}{N}\right) + v_k^2(N-1)\sin^2\left(\frac{2\pi k}{N}\right) \\
&= v_k^2(N) - 2v_k(N)v_k(N-1)\cos\left(\frac{2\pi k}{N}\right) \\
&\quad + v_k^2(N-1)\left[\cos^2\left(\frac{2\pi k}{N}\right) + \sin^2\left(\frac{2\pi k}{N}\right)\right] \\
&= v_k^2(N) + v_k^2(N-1) - 2\cos\left(\frac{2\pi k}{N}\right)v_k(N)v_k(N-1)
\end{aligned}
\tag{8.58}
$$

(b) 对于数字“9”的DTMF音调是1477 Hz和852 Hz。每一个音调的检测要求一对Goertzel IIR滤波器。对于1477音调，对应的频率单元是38和74：

$$
a_1 = 2\cos\left(\frac{2\pi\times 38}{205}\right) = 0.79;\quad a_2 = -1
$$

$$
a_1' = 2\cos\left(\frac{2\pi\times 74}{210}\right) = -1.1996;\quad a_2' = -1
$$

对于852 Hz音调，系数是

$$
a_1 = 1.5623,\ a_2 = -1;\qquad a_1' = 0.5,\ a_2' = -1
$$

8.19.3 数据通信的时钟恢复

在大多数的长距离数字数据通信中，一个基本问题是在接收端以正确的频率和相位产生一个时钟，这样数据可以被正确解码。时钟通常是从接收到的数据中推导出来的。

传统上是利用模拟电路（例如利用锁相环）来进行时钟恢复，但是它们易受时间和温度飘移的影响。另外，这样的电路在包含短脉冲串传输的应用中是不合适的，因为它们响应慢，对于那些超过一个数据率的应用也是不合适的（Smithson, 1992）。

输入数据流通常是在发射端被量化（在空闲周期里提供时钟信息），然后被编码，每一个码代表一个符号。然后码以所谓的符号率发射。在接收端的问题就是恢复符号时钟。

图8.41给出了利用DSP的符号时钟恢复的原理。数据流延时半个时钟周期后和它自身做模2相加（即异或），产生一个输出（在点C处），它包含依符号率改变的电平。接着将数据加到一个边缘稳定的带通IIR滤波器。这样的一个滤波器的冲激响应随时间衰减得非常慢，在滤波器的中心频率w_0处产生一个“阻尼振荡”。边缘稳定滤波器的应用确保有一个输出，即使在一个合理长度的时间内输入数据流中没有传输。滤波器的抽样频率选择为符号率的倍数。期望的符号时钟是利用过零检测（图8.41的点E处）从滤波器的输出中推导出来的。对于2的补码表示，这可以通过检验数字滤波器输出端数据样本的符号而轻易实现。

图8.42所示的简单的全极点IIR滤波器可以用来做符号时钟恢复，滤波器的特性由下面的传递函数给定：

$$
H(z) = \frac{1}{[z - r\exp(-jw_0T)][z - r\exp(jw_0T)]} = \frac{1}{z^2 - 2r\cos(w_0T)z - r^2}
$$

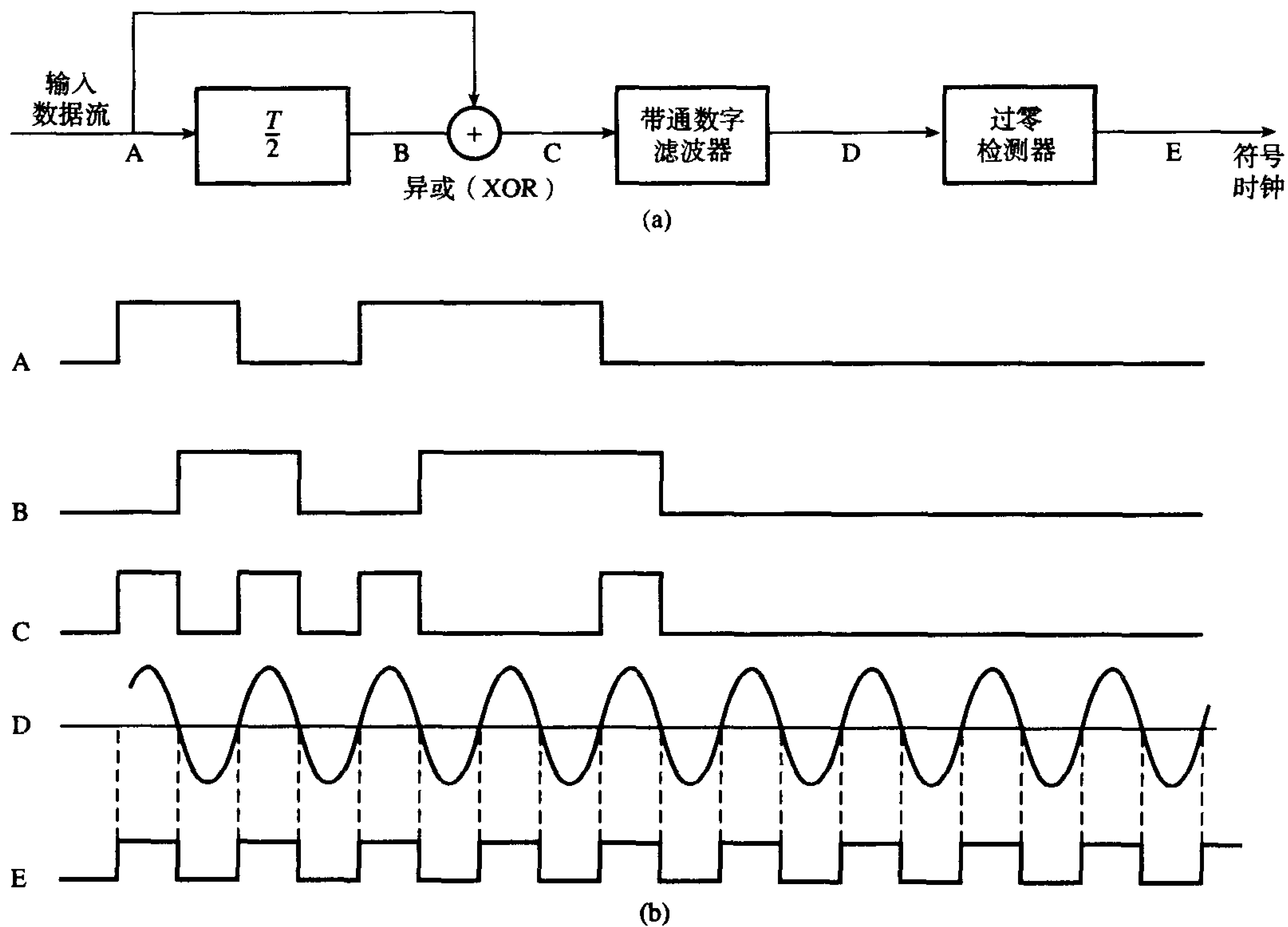


图 8.41　数据通信中符号时钟恢复的原理图

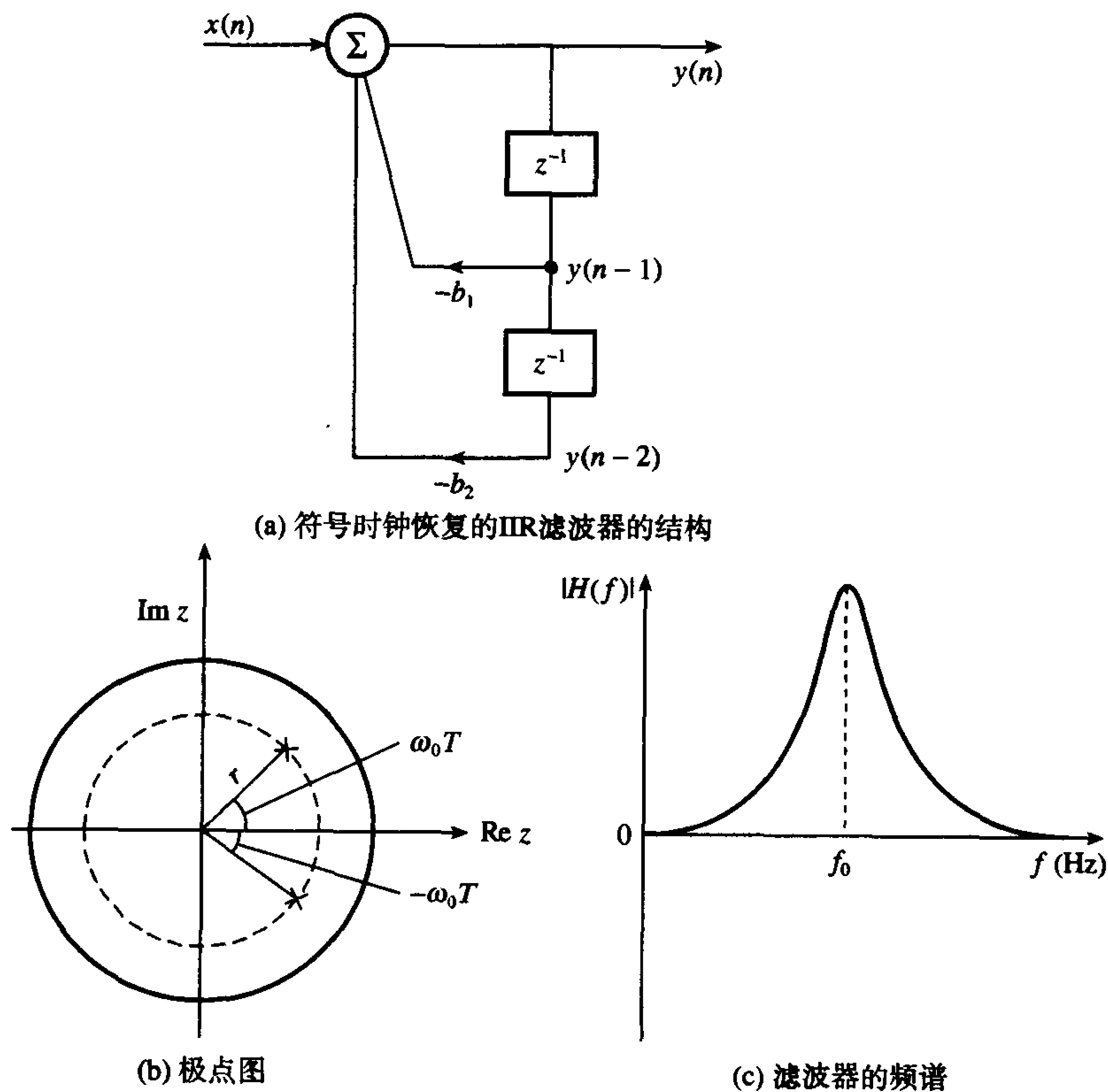


图 8.42　简单的全极点 IIR 滤波器

其中 w_0 是带通滤波器的中心频率，r 是极点的半径，T 是抽样频率的倒数，w_0 通常选择为等于或非常接近待恢复的符号时钟频率，抽样频率是中心频率的倍数。滤波器的带宽是由极点的半径决定的（参见8.4式）。为了确保冲激响应是随时间缓慢衰减，极点通常放置于非常靠近单位圆的地方，通常是在 $0.99 < r < 1$ 的范围内。如8.5.1节里讨论的那样（8.5式），极点半径 r 和滤波器带宽 bw 有如下关系：

$$r \approx 1 - (\mathrm{bw}/F_s)\pi$$

其中 $F_s = 1/T$ 是滤波器的抽样频率。

例如，为了恢复一个假设为4800波特的调制解调器的符号时钟，合适的滤波器参数是

数据率	4.8 k 波特
滤波器中心频率 f_0	4.8 kHz
抽样频率	153.6 kHz
通带宽度，bw	100 Hz

在这种情况下，极点半径（从上面的方程可得）$r = 0.997\,954\,69$，极点角度是 $w_0T = 2\pi f_0 T = (2\pi \times 4.8 \times 10^3/153.6 \times 10^3) = 0.196\,35$ 弧度 $\approx 11.25°$。最后得到的传递函数是

$$H(z) = \frac{1}{z^2 - 1.957\,558z + 0.995\,913}$$

如本章开始讨论的那样，如果滤波器要如期望的那样工作，有限字长效应必须考虑。特别是，滤波器的输入需要进行伸缩变换，以避免在它的输出端因为溢出而自激，而应用简单的舍入噪声整形方法，可以帮助产生一个“干净”的时钟。在一个实际的时钟恢复系统中，当输入数据是1或者0序列时，为了改善系统的性能，必须要有第二级滤波器（Smithson, 1992）。

习题

8.1 一个低通滤波器具有如下所示位置的极点和零点：

零点，-0.5；极点，$0.370, 0.6 \pm 0.5\mathrm{j}$

(1) 画出极零图。

(2) 求传递函数 $H(z)$。

8.2 利用冲激不变法，对具有下面传递函数的模拟滤波器进行数字化：

$$H(s) = \frac{\alpha}{s(s+\alpha)}, \quad \alpha = 0.5$$

假设抽样频率是1（归一化的）。

8.3 要求在一个数字计算机里模拟一个具有如下归一化特性的模拟系统：

$$H(s) = \frac{1}{s^2 + \sqrt{2}s + 1}$$

利用下列方法求合适的传递函数，

(1) 冲激不变法。

(2) 双线性变换法。

假设抽样频率是5 kHz，3 dB 的截止频率是1 kHz。

8.4 利用BZT法，确定图8.43所示的电阻－电容（RC）滤波器的数字等效形式的传递函数和差分方程。假设抽样频率是150 Hz，截止频率是30 Hz。

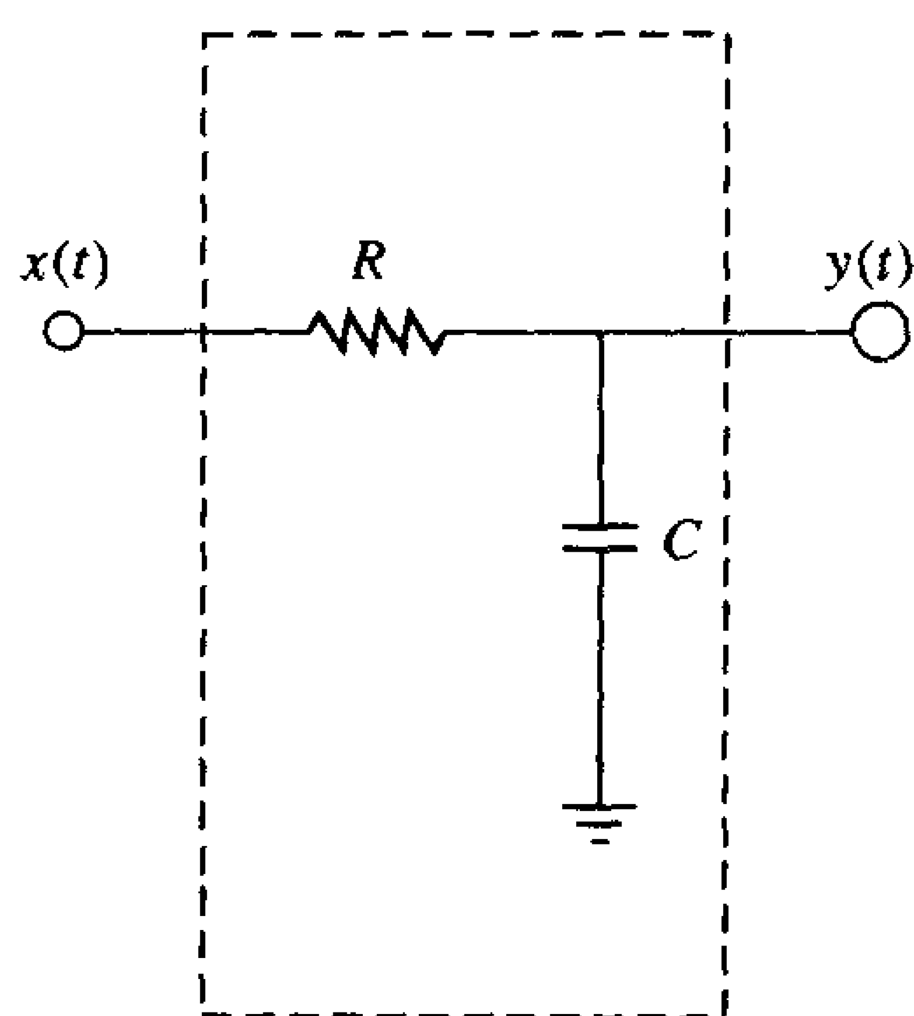


图 8.43　习题 8.4 中的电阻－电容滤波器的等效电路图

8.5　求满足下面性能规范的IIR数字带通滤波器。从一个合适的归一化模拟LPF出发，(i)利用合适的频带变换和BZT，求因式形式的$H(z)$的系数，(ii)画出模拟滤波器以及最后得到的数字带通滤波器的。

通带	8 ~ 10 kHz
抽样频率	32 kHz
带通滤波器的阶数	4
滤波器类型	巴特沃斯

8.6　(a) 在应用双线性z变换（BZT）设计法时，要被数字化的滤波器在靠近奈奎斯特频率时，其响应会出现“扭曲效应”。解释这个效应在实践中的意义，给出一个特定应用例子。

(b) 需要设计一个 IIR 高通滤波器，它具有巴特沃斯特性，满足下面的性能规范：

通带	2 ~ 4 kHz
阻带	0 ~ 500 Hz
通带波纹	3 dB
阻带衰减	20 dB
抽样频率	8 kHz

(i)　求一个合适的模拟原型低通滤波器的通带和阻带边沿频率。你的答案必须包括如何从性能规范得出原型低通滤波器的基本概念的细节；

(ii)　求原型低通滤波器的阶数N；

(iii) 利用 BZT 法，求出传递函数，即 IIR 滤波器的系数。

8.7　设计一个数字带通滤波器，它具有巴特沃斯特性，满足下面的性能规范：

通带	200 ~ 300 Hz
抽样频率	2000 Hz
滤波器阶数	2

(a) (i)　从合适的原型模拟低通滤波器出发，利用双线性z变换法确定数字滤波器的传递函数。

(ii) 利用极零图，解释如何连续地把原型滤波器的极点和零点从s平面映射到z平面。

(b) 假设(a)中的滤波器用定点算法执行。评估一下把系数量化到8位对极点位置和中心频率的影响。

注意：从低通到带通的转换关系为

$$s \to \frac{s^2 + \omega_0^2}{Bs}$$

其中

$$\omega_0^2 = \omega_1'\omega_2'; \qquad B = \omega_2' - \omega_1'$$

ω_1'、ω_2' 和 ω_0 分别是上、下带沿频率和中心频率。

8.8 利用双线性变换法获得一个适当的IIR数字滤波器的系数，该滤波器具有椭圆特性，满足下面的性能规范：

通带	4 ~ 12 kHz
阻带	0 ~ 3.4 kHz
	12.6 ~ 16 kHz
通带波纹	< 0.1 dB
阻带衰减	> 30 dB
抽样频率	32 kHz

确定一个合适的系数字长以确保滤波器的稳定以及频率响应在规定的限制之内。

8.9 用软件设计和实现一个数字低通滤波器，要求满足下面的性能规范：

通带边沿	2.5 kHz
阻带边沿	3 kHz
通带偏差	< 0.1 dB
阻带衰减	> 60 dB
抽样频率	15 kHz

8.10 通过双线性变换，求数字滤波器的系数。该滤波器在通带 0 ~ 4 kHz 内最为平坦，在超过 10 kHz 的频率至少有 25 dB 的衰减。假设抽样频率是 32 kHz。

8.11 某个低通滤波器的要求如下：

通带	0 ~ 30 Hz
阻带边沿	50 Hz
阻带衰减	> 40 dB
	在 f > 50 Hz 处
抽样频率	256 Hz

假设滤波器具有巴特沃斯响应，通过双线性变换求滤波器的传递函数 $H(z)$。求出利用二阶或一阶部分串联形式的实现结构图。

8.12 某个实时数字信号处理系统需要设计一个具有巴特沃斯响应的带通数字滤波器。这个滤波器的要求是

通带	0.3 ~ 3.4 kHz
阻带	0 ~ 0.2 kHz 和 4 ~ 8 kHz

阻带衰减	25 dB
抽样频率	32 kHz

利用双线性变换法求滤波器的传递函数。

8.13 在某些生物医学应用中，需要用一个数字滤波器来消除因为身体运动而引起的基线漂移和伪像。这个滤波器要满足下面的要求：

通带	1 ~ 30 Hz
阻带	0 ~ 0.5 Hz 和 40 ~ 128 Hz
通带波纹	< 0.1 dB
阻带衰减	> 30 dB
抽样频率	256 Hz

求合适的 IIR 滤波器的阶数以及它的传递函数 $H(z)$。

8.14 设计一个窄带抑制滤波器用以移去干扰信号。滤波器应该满足如下性能规范：

通带边沿	45 Hz 和 55 Hz
通带波纹	< 0.1 dB
阻带衰减	> 50 dB
抽样频率	500 Hz

求滤波器的系数。

8.15 利用冲激不变法，确定一个数字滤波器的传递函数和差分方程，该数字滤波器和一个单极点 RC 低通滤波器等价。假设抽样频率是 150 Hz，3 dB 的截止频率是 30 Hz。

8.16 具有简单极点的标准二阶模拟滤波器部分，可以表示成如下形式：

$$\frac{A_0 + A_1 s}{B_0 + B_1 s + B_2 s^2} = \frac{C_1}{s + p_1} + \frac{C_2}{s + p_2}$$

其中 C_1 和 C_2 是部分因式的系数，p_1 和 p_2 是 s 平面的极点。假设给出的二阶部分的冲激不变转换是

$$\frac{A_0 + A_1 s}{B_0 + B_1 s + B_2 s^2}$$
$$\to \frac{c_1 + c_2 - (c_1 \mathrm{e}^{-p_2 T} + c_2 p^{-p_1 T}) z^{-1}}{1 - (\mathrm{e}^{-p_1 T} + \mathrm{e}^{-p_2 T}) z^{-1} + \mathrm{e}^{-(p_1 + p_2) T} z^{-2}}$$
$$= \frac{a_0 - a_1 z^{-1}}{1 + b_1 z^{-1} + b_2 z^{-2}}$$

其中 T 是抽样间隔。

(1) 求用 A_0、A_1、B_0、B_1 和 B_2 表示的 p_1、p_2、C_1 和 C_2。

(2) 当极点是复共轭对的时候，求出系数 a_0、a_1、b_1 和 b_2 的表达式。

(3) 在极点是实数且不相等的情况下，重复(2)。

(4) 给定下列归一化模拟传递函数，利用你的结果来求等价的离散滤波器的系数。假设抽样频率是 10 kHz，截止频率是 2 kHz。

$$H(s) = \frac{1}{s^2 + \sqrt{2} s + 1}$$

8.17 从合适的模拟切比雪夫 LPF 出发，求满足下面性能规范的数字带阻滤波器的传递函数：

阻带	10 ~ 15 kHz
抽样频率	50 kHz
通带波纹	0.5 dB
滤波器阶数	6

8.18 从一个合适的模拟切比雪夫 LPF 和如下式给定的双二次变换出发，利用 s 平面到 z 平面映射的方法，求满足下面性能规范的数字带通滤波器的传递函数：

通带	10 ~ 15 kHz
抽样频率	50 kHz
通带波纹	0.5 dB
带通滤波器的阶数	6

对带通和带阻滤波器来说，BZT 的一个替代方法是下面的双二次转换（Gold and Rader, 1969; Gray and Markel, 1976）：

$$s = \cot\left[\frac{(\omega_2 - \omega_1)T}{2}\right]\left[\frac{z^2 - 2z\cos\gamma + 1}{z^2 - 1}\right]$$

低通到带通

$$s = \tan\left[\frac{(\omega_2 - \omega_1)T}{2}\right]\left[\frac{z^2 - 1}{z^2\ 2z\cos\gamma + 1}\right]$$

低通到带阻

其中

$$\cos\gamma = \cos\left[\frac{(\omega_2 + \omega_1)T}{2}\right]\Big/\cos\left[\frac{(\omega_2 - \omega_1)T}{2}\right]$$

ω_1 和 ω_2 是分别是下、上带沿频率（对于 BPF 是通带边沿频率，对于 BSF 是阻带边沿频率），γ 是中心频率。

8.19 一个模拟低通滤波器的特性由一对位于 s 平面的极点 $p_{1,2} = -1.4 \pm 1.2936\mathrm{j}$ 给定。希望把它转化成通带边沿为 3 kHz 的带通数字滤波器，抽样频率是 15 kHz。给定数字低通到带通的转换和 8.40 式的 BZT。

(1) 求数字带通滤波器的极点和零点。

(2) 求它的因式形式的传递函数。

8.20 数字电话的双音多频（DTMF）检测方法是利用一系列二阶 Goertzel 滤波器来提取出 DTMF 音调和它们的二次谐波。假设对数字“0”的 DTMF 音调是 941 Hz 和 1336 Hz。对于基频和二次谐波，序列长度取 $N = 205$ 和 210，对应的离散频率单元是 24 和 47，求低频音调 Goertzel 滤波器的反馈回路的系数值。

8.21 (a) 借助框图，解释采用 Goertzel 算法的按钮式电话的双音多频（DTMF）检测方法的原理。

(i) 数据序列 $x(n)$（$n = 0, 1, \ldots, N-1$）的离散傅里叶变换（DFT）定义为

$$X(k) = \sum_{m=0}^{N-1} x(m) W_N^{km},$$
$$k = 0, 1, \ldots, N-1$$

其中 W_N^{km} 是旋转因子。

(ii) 从上面的等式出发，证明 DTMF 音调检测的 Goertzel 滤波器的 z 平面传递函数 $H_k(z)$可以用递归形式表示为

$$H_k(z) = \frac{1 - W_N^k z^{-1}}{1 - 2\cos\left(\dfrac{2\pi k}{N}\right) z^{-1} + z^{-2}}$$

(iii) 推导 Goertzel 滤波器输出的幅度平方 $|y_k(n)|^2$ 在离散时刻 $n = N$ 时的表达式，并且证明在修正 Goertzel 算法中不要求复数运算。

(b) 按钮式电话系统的 DTMF 音调检测方法是基于表 8.3 里总结的性能规范以及二阶 Goertzel滤波器。如果拨了数字“00”，计算接收端解码数字的Goertzel滤波器的系数。对DTMF音调进行检测，比较Goertzel算法和以基-2 FFT 算法的计算复杂度，并证明在这个应用里不用 FFT 算法而用 Goertzel 算法的合理性。阐述任何合理的假设。

MATLAB 习题

8.22 设计一个具有切比雪夫特性的带通数字滤波器，它满足下面的性能规范：

通带	1200 ~ 1800 Hz
阻带衰减	> 30 dB
通带波纹	< 0.5 dB
过渡带宽	400 Hz
抽样频率	7.5 kHz

采用合适的软件程序，求滤波器的系数。

8.23 一个模拟滤波器要转换成等价的数字滤波器，数字滤波器以 256 Hz 的抽样率工作。假设模拟滤波器具有下面的传递函数：

$$H(s) = \frac{1}{s^3 + 2s^2 + 2s + 1}$$

(1) 求数字滤波器的合适系数。

(2) 假设这个数字滤波器要用串联结构实现，画出的实现的框图，并建立差分方程。

(3) 对并联结构重复(2)。

8.24 一个 IIR 滤波器具有如下的传递函数：

$$H(z) = \frac{0.1436 + 0.2872z^{-1} + 0.1436z^{-2}}{1 - 1.8353z^{-1} + 0.9748z^{-2}}$$

(1) 确定极点和零点的位置，并画出极零图。

(2) 求出极点到原点的径向距离。

(3) 估计表示每一个系数所需要的位数，

(a) 为了保持稳定；

(b) 为了使通带里的幅度响应的变化不超过 1%。

8.25 从如下几方面比较匹配 z 变换法、冲激不变法和双线性 z 变换法，

(a) 奈奎斯特效应对幅度频率、相位和群延迟响应的影响。

(b) 极零图的分布。

在你的研究中利用 MATLAB 和下面的滤波器。

(1) 低通滤波器

一个椭圆的低通数字滤波器具有如下的性能规范：

通带	0 ~ 1 kHz
阻带	3 ~ 5 kHz
通带波纹	1 dB
阻带衰减	60 dB
抽样频率	10 kHz

(2) 高通滤波器

一个椭圆的高通数字滤波器具有下面的性能规范：

阻带	0 ~ 1 kHz
通带	3 ~ 5 kHz
通带波纹	1 dB
阻带衰减	60 dB
抽样频率	10 kHz

(3) 带通滤波器

(a) 一个巴特沃斯带通滤波器具有下面的性能规范：

通带	200 ~ 300Hz
抽样频率	2000 Hz
滤波器阶数	8

(b) 一个巴特沃斯带通数字滤波器具有下面的性能规范：

通带	800 ~ 900 Hz
抽样频率	2000 Hz
滤波器阶数	8

(4) 带阻滤波器

一个椭圆带阻数字滤波器具有下面的性能规范：

通带	0 ~ 15 kHz
	30 ~ 50 kHz
阻带	20 ~ 25 kHz
通带波纹	0.2 dB
阻带衰减	40 dB
抽样频率	100 kHz

8.26 从以下几个方面比较数字巴特沃斯、切比雪夫类型 I、切比雪夫类型 II 和椭圆滤波器的性质：

(a) 它们极点和零点的分布；
(b) 滤波器阶数；
(c) 幅度 - 频率响应的过渡带宽，通带和阻带波纹；
(d) 相位和群延迟响应。

在你的研究中利用如下的滤波器：

(1) 低通滤波器

通带	0 ~ 500 Hz
阻带	2 ~ 4 kHz
通带波纹	3 dB
阻带衰减	20 dB
抽样频率	8 kHz

(2) 高通滤波器

阻带	0 ~ 500 Hz
通带	2 ~ 4 kHz
通带波纹	3 dB
阻带衰减	20 dB
抽样频率	8 kHz

(3) 带通滤波器

(a) 带通下沿频率	250 Hz
带通上沿频率	300 Hz
阻带下沿频率	50 Hz
阻带上沿频率	450 Hz
通带波纹	3 dB
阻带衰减	20 dB
抽样频率	1 kHz
(b) 通带	0 ~ 15 kHz
	30 ~ 50 kHz
阻带	20 ~ 25 kHz
通带波纹	0.2 dB
阻带衰减	40 dB
抽样频率	100 kHz

(4) 带阻滤波器

(a) 通带	0 ~ 50 Hz
	450 ~ 500 Hz
阻带	250 ~ 300 Hz
通带波纹	3 dB
阻带衰减	20 dB
抽样频率	1 kHz

(b) 通带	0 ~ 15 kHz
	30 ~ 50 kHz
阻带	20-25 kHz
通带波纹	0.2 dB
阻带衰减	40 dB
抽样频率	100 kHz

8.27 (a) (i) 利用BZT法和MATLAB，求一个数字混频器里用来做音频信号处理的离散钟形滤波器的传递函数，数字混频器中控制设置对应于Q因子是2，在10 kHz处有6.02 dB的提升（峰值）。假设抽样频率是48 kHz，等价的模拟滤波器的s平面传递函数如下给定。

画出离散滤波器的幅度、相位、群延迟响应以及极零图。

(ii) 假设抽样频率是96 kHz，重做上面的问题。

(b) 利用MZT法重做(a) (i)和(ii)。

(c) 利用冲激不变法重做(a) (i)和(ii)。

(d) 比较(a)和(c)的结果。

$$H(s) = \frac{s^2 + (3+k)\dfrac{\omega_0}{Q}s + \omega_0^2}{s^2 + (3-k)\dfrac{\omega_0}{Q}s + \omega_0^2}$$

其中

$$k = 3\left(\frac{G-1}{G+1}\right); \quad \omega_0 = \text{提升频率}$$

$$G = \text{增益因子}; \quad Q = \text{Q因子}$$

8.28 (a) 开发一个检测DTMF音调和其二次谐波的C语言伪代码。阐述所做的任何假设。

(b) 利用MATLAB重做(a)。

(c) 利用MATLAB，产生合适的频率音调，并利用这些来验证用来检测DTMF音调的MATLAB程序。

参考文献

Abu-el-Haija A. and Al-Ibrahim M.M. (1986) Improving performance of digital sinusoidal oscillators by means of error feedback circuits. *IEEE Trans. Circuits and Systems*, **33**(4), 373–80.

Antoniou A. (1979) *Digital Filters Analysis and Design*. New York: McGraw-Hill.

Chen C.J. (1996) *Modified Goertzel Algorithm in DTMF Detection using the TMS320C80*, Texas Instruments, Application Report SPRA066, June.

Clark R.J., Ifeachor E.C. and Rogers G.M. (1996) Real-time equaliser coefficient realisation with minimised computational load and distortion. Preprint 4360, 101st Audio Engineering Society Convention.

Clark R.J., Ifeachor E.C., Rogers G.M. and Van Eetvelt P.W.J. (2000) Techniques for generating digital equaliser coefficients. *J. Audio Engineering Society*, **48**(4), 281–98.

Dattorro J. (1988) The implementation of recursive digital filters for high-fidelity audio. *J. Audio Engineering Society*, **36**(11), 851–78.

DeFatta D.J., Lucas J.G. and Hodgkiss W.S. (1988) *Digital Signal Processing*. New York: Wiley.

Feeney S.L., Kieburtz R.B., Mina K.V. and Tewksbury S.K. (1971) Design of digital filters for an all digital frequency division multiplex–time division multiplex translator. *IEEE Trans. Circuit Theory*, **18**, 702–11.

Gold B. and Rader C.M. (1969) *Digital Processing of Signals*. New York: McGraw-Hill.

Gray A.H. and Markel J.D. (1976) A computer program for designing elliptic filters. *IEEE Trans. Acoustics, Speech and Signal Processing*, **24**(6), 529–38.

IEEE (1979) *Programs for Digital Signal Processing*. New York: IEEE Press.
Jackson L.B. (1986) *Digital Filters and Signal Processing*. Boston MA: Kluwer.
Jackson L.B., Kaiser J.F. and McDonald H.S. (1968) An approach to the implementation of digital filters. *IEEE Trans. Audio and Electroacoustics*, **16**(3), 413–21.
Jong M.T. (1982) *Methods of Discrete Signal System Analysis*. New York: McGraw-Hill.
Lynn P.A. and Fuerst W. (1989) *Introductory Digital Signal Processing with Computer Applications*. Chichester: Wiley.
Marven C. (1990) General-purpose tone decoding and DTMF detection, in *Theory, Algorithms, and Implementations, Digital Signal Processing Applications with the TMS320 Family*, Vol. 2, literature number SPRA016, Texas Instruments.
Mock P. (1985) Add DTMF generation and decoding to DSP-μp designs. *EDN*, **30**.
Parks T.W. and Burrus C.S. (1987) *Digital Filter Design*. New York: Wiley.
Rabiner L.R. and Gold B. (1975) *Theory and Applications of Digital Signal Processing*. Englewood Cliffs NJ: Prentice-Hall.
Rader C.M. and Gold B. (1967) Effects of parameter quantization on the poles of a digital filter. *Proc. IEEE*, **55**, 688–9.
Smithson P. (1992) Clock recovery for a satellite data modem, University of Plymouth (personal communication).
Stanley W.D., Dougherty G.R. and Dougherty R. (1984) *Digital Signal Processing*, 2nd edn. Reston VA: Reston Publishing, Inc.
Stearns S.D. and Hush D.R. (1990) *Digital Signal Analysis*, 2nd edn. Englewood Cliffs NJ: Prentice-Hall.

参考书目

Abu-el-Haija A.I. and Peterson A.M. (1979) An approach to eliminate roundoff errors in digital filters. *IEEE Trans. Acoustics, Speech and Signal Processing*, **27**, 195–8.
Ahmed N. and Natarajan T. (1983) *Discrete-time Signals and Systems*. Reston VA: Reston Publishing, Inc.
Allen J. (1975) Computer architecture for signal processing. *Proc. IEEE*, **63**(4), 624–48.
Arjmand M. and Roberts R.A. (1981) On comparing hardware implementations of fixed point digital filters. *IEEE Circuits Systems Mag.*, **3**(2), 2–8.
Avenhaus E. (1972) Filters with coefficients of limited wordlength. *IEEE Trans. Audio Electroacoustics*, **20**, 206–12.
Barnes C.W., Tran B.N. and Leung S.H. (1985) On the statistics of fixed-point roundoff error. *IEEE Trans. Acoustics, Speech and Signal Processing*, **33**, 595–606.
Bellanger M. (1984) *Digital Processing of Signals. Theory and Practice*. New York: Wiley.
Chang T.L. (1978) A low roundoff noise digital filter structure. In *Proc. IEEE Int. Symp. on Circuits and Systems*, May 1978, pp. 1004–8.
Chang T.L. (1979) Error-feedback digital filters. *Electronics Lett.*, **15**, 348–9.
Chang T.L. (1980) Comments on 'An approach to eliminate roundoff errors in digital filters'. *IEEE Trans. Acoustics, Speech and Signal Processing*, **28**(2), 244–5.
Chang T.L. (1981) Suppression of limit cycles in digital filters designed with one magnitude-truncation quantizer. *IEEE Trans. Circuits and Systems*, **28**(2), 107–11.
Chang T.L. (1981) On low-roundoff noise and low-sensitivity digital filter structures. *IEEE Trans. Acoustics, Speech and Signal Processing*, **29**(5), 1077–80.
Chang T.L. and White S.A. (1981) An error cancellation digital-filter structure and its distributed-arithmetic implementation. *IEEE Trans. Circuits and Systems*, **28**(4), 339–42.
Charalambous C. and Best M.J. (1974) Optimization of recursive digital filters with finite wordlengths. *IEEE Trans. Acoustics, Speech and Signal Processing*, **22**(6), 424–31.
Chassaing R. and Horning D.W. (1990) *Digital Signal Processing with the TMS320C25*. New York: Wiley.
Claasen T.A.C.M. and Kristiansson L.O.G. (1975) Necessary and sufficient conditions for the absence of overflow phenomena in a second order recursive digital filter. *IEEE Trans. Acoustics, Speech and Signal Processing*, **23**(6), 509–15.
Claasen T.A.C.M., Mecklenbrauker W.F.G. and Peek J.B.H. (1973) Second-order digital filter with only one magnitude-truncation quantiser and having practically no limit cycles. *Electronics Lett.*, **9**, 531–2.
Claasen T.A.C.M., Mecklenbrauker W.F.G. and Peek J.B.H. (1973) Some remarks on the classification of limit cycles in digital filters. *Philips Research Rep.*, **28**, 297–305.
Claasen T., Mecklenbrauker W.F.G. and Peek J.B.H. (1975) Frequency domain criteria for the absence of zero-input limit cycles in nonlinear discrete-time systems, with applications to digital filters. *IEEE Trans. Circuits and Systems*, **22**, 232–9.
Claasen T.A.C.M., Mecklenbrauker W.F.G. and Peek J.B.H. (1976) Effects of quantization and overflow in recursive digital filters. *IEEE Trans. Acoustics, Speech and Signal Processing*, **24**(6), 517–28.
Clark R.J., Ifeachor E.C., Rogers G.M. and Van Eetvelt P.W.J. (2000) Techniques for generating digital equaliser coefficients. *Journal of Audio Engineering Society*, **48**(4), 281–98.

Crochiere R.E. (1975) A new statistical approach to the coefficient wordlength problem for digital filters. *IEEE Trans. Circuits and Systems*, **22**, 190–6.

Crochiere R.E. and Oppenheim A.V. (1975) Analysis of linear digital networks. *Proc. IEEE*, **63**(4), 581–94.

Diniz P.S.R. and Antoniou A. (1985) Low-sensitivity digital filter structures which are amenable to error-spectrum shaping. *IEEE Trans. Circuits and Systems*, **32**(10), 1000–7.

Elliot D.F. (ed.) (1987) *Handbook of Digital Signal Processing*. London: Academic Press.

IEEE (1978) Digital Signal Processing II. Institute of Electrical and Electronics Engineers.

Jackson L.B. (1970) On the interaction of roundoff noise and dynamic range in digital filters. *BSTJ*, **49**(2), 159–84.

Jackson L.B. (1976) Roundoff noise bounds derived from coefficient sensitivities for digital filters. *IEEE Trans. Circuits and Systems*, **23**(8), 481–5.

Knowles J.B. and Olcayto E.M. (1968) Coefficient accuracy and digital filter response. *IEEE Trans. Circuit Theory*, **15**, 31–41.

Liu B. (1971) Effect of finite wordlength on the accuracy of digital filters – a review. *IEEE Trans. Circuit Theory*, **18**, 670–7.

Liu B. and Kaneko T. (1969) Error analysis of digital filters realized with floating-point arithmetic. *Proc. IEEE*, **57**(10), 1735–47.

Markel J.D. and Gray A.H. (1975) Fixed-point implementation algorithms for a class of orthogonal polynomial filter structures. *IEEE Trans. Acoustics, Speech and Signal Processing*, **23**(5), 486–94.

Markel J.D. and Gray A.H. (1975) Roundoff noise characteristics of a class of orthogonal polynomial structures. *IEEE Trans. Acoustics, Speech and Signal Processing*, **23**(5), 473–86.

Motorola (1988) *Digital Stereo 10-band Graphic Equalizer Using the DSP56001*. Motorola Application Note.

Mullis C.T. and Roberts R.A. (1976) Round-off noise in digital filters: frequency transformations and invariants. *IEEE Trans. Acoustics, Speech and Signal Processing*, **24**(6), 538–50.

Munson D.C. and Liu B. (1980) Low-noise realization for narrow-band recursive digital filters. *IEEE Trans. Acoustics, Speech and Signal Processing*, **28**, 41–54.

Nagle H.T. and Nelson V.P. (1981) Digital filter implementation on 16 bit microcomputers. *IEEE Micro*, **1**, 23–41.

Oppenheim A.V. and Schafer R.W. (1975) *Digital Signal Processing*. Englewood Cliffs NJ: Prentice-Hall.

Oppenheim A.V. and Weinstein, C.J. (1972) Effects of finite register length in digital filtering and the fast Fourier transform. *Proc. IEEE*, **60**, 957–76.

Peled A., Liu B. and Steiglitz K. (1974) A new hardware realization of digital filters. *IEEE Trans. Acoustics, Speech and Signal Processing*, **22**, 456–62.

Peled A., Liu B. and Steiglitz K. (1975) A note on implementation of digital filters. *IEEE Trans. Acoustics, Speech and Signal Processing*, **23**, 387–9.

Rabiner L.R., Cooley J.W., Helms H.D., Jackson L.B., Kaiser J.F., Rader C.M., Schafer R.W., Steiglitz K. and Weinstein C.J. (1972) Terminology in digital signal processing. *IEEE Trans. Audio and Electroacoustics*, **20**, 322–37.

Sandberg I.W. and Kaiser J.F. (1972) A bound on limit cycles in fixed-point implementations of digital filters. *IEEE Trans. Audio and Electroacoustics*, **20**, 110–12.

Schmalzel J.L., Heine D.N. and Ahmed N. (1980) Some pedagogical considerations of digital filter hardware implementation. *IEEE Circuits and Systems Mag.*, **2**(1), 4–13.

Sim P.K. and Pang K.K. (1985) Effects of input-scaling on the asymptotic overflow-stability properties of second recursive digital filters. *IEEE Trans. Circuits and Systems*, **32**(10), 1008–15.

Steiglitz K. (1971) Designing short-word recursive digital. *Proc. 9th Ann. Allerton Conf. on Circuit and System Theory*, 6–8 October, 1971, pp. 778–88.

Steiglitz K., Bede L. and Liu B. (1976) An improved algorithm for ordering poles and zeros of fixed-point recursive digital filters. *IEEE Trans. Acoustics, Speech and Signal Processing*, **24**, 341–3.

Taylor F.J. (1983) *Digital Filter Design Handbook*. New York: Marcel Dekker.

Thong T. (1976) Finite wordlength effects in the ROM digital filter. *IEEE Trans. Acoustics, Speech and Signal Processing*, **24**, 436–7.

Thong T. and Liu B. (1977) Error spectrum shaping in narrowband recursive digital filters. *IEEE Trans. Acoustics, Speech and Signal Processing*, **25**, 200–3.

Williamson D. and Sridharan S. (1985) An approach to coefficient wordlength reduction in digital filters. *IEEE Trans. Circuits and Systems*, **32**(9), 893–903.

附录

8A　IIR 数字滤波器设计的 C 程序

我们为本书开发了设计 IIR 数字滤波器的 C 语言程序。由于受篇幅的限制，这里只列出系数计算的冲激不变法的程序。下列程序在书中没有列出，不过在指导手册的 CD 中可以找到。网上也给出了 MATLAB m 文件，它们可以用来执行和这几个 C 程序相似的任务（详情请参见前言）。

- 双线性变换；
- 利用双线性变换法的经典 IIR 滤波器（巴特沃斯、切比雪夫和椭圆）的系数计算；
- IIR 滤波器的有限字长分析。

8A.1　冲激不变法的 C 程序

请注意在 C 程序里，模拟传递函数的系数 A_k 和 B_k 和离散时间滤波器传递函数的系数 a_k 和 b_k，分别是分子和分母的系数矢量。和 C 程序不同，在 8A.1 式到 8A.8 式里我们修正了这些系数的用处，使得它们和 MATLAB 程序里的一致。

在程序 8A.1 里，我们列出了冲激不变法的程序。首先我们对程序所基于的概念进行了总结，接着用一个例子来解释它的应用。这个讨论是基于二阶滤波器部分的，因为这是数字 IIR 滤波器的基本构件。

考虑一个一般的二阶 s 平面传递函数，它要被转换成一个离散传递函数：

$$H(s)=\frac{B_0+B_1s}{A_0+A_1s+A_2s^2} \tag{8A.1}$$

等价的 z 平面传递函数也是一个具有如下形式的二阶部分：

$$H(z)=\frac{b_0+b_1z^{-1}}{1+a_1z^{-1}+a_2z^{-2}} \tag{8A.2}$$

程序 8A.1　冲激不变法（注意：在这个程序里，系数 A 和 B 以及 a 和 b，和上面方程里是相反的）

```
/*-------------------------------------------------------------------------*
 *      Impulse invariant method                                           *
 *                                                                         *
 *      The analog transfer function must be frequency-scaled              *
 *      (normalized frequency) before using program                        *
 *      30.10.92                                                           *
 *-------------------------------------------------------------------------*
 */
#include    <stdio.h>
#include    <math.h>
#include    <dos.h>

void        dfilter();
double      T;
double      a0, a1, a2, b0, b1, b2;
double      p1, p2, pr, pi;
double      c1, c2, cr, ci;
```

```
float       A0, A1, B0, B1, B2, temp;

main()
{
                                                    /* initialize coeffs */
            A0=0; A1=0; B0=1; B1=0; B2=0;
            a0=0; a1=0; b0=1; b1=0; b2=0;
            c1=0; c2=0; p1=0; p2=0; a2=0;
                                                    /* read s-plane coefficients */
            printf("impulse invariant discrete filters \n");
            printf("\n");
            printf("enter s-plane coefficients \n");
            printf("enter denominator coeffs: B0, B1, B2 \n");
            scanf("%f %f %f", &B0, &B1, &B2);
            printf("enter numerator coeffs: A0, A1 \n");
            scanf("%f %f", &A0, &A1);
            T=1;
            dfilter();
            printf("\n");
            printf("press enter to continue\n");
            getch();
            exit(0);
}
/*---------------------------------------------------------------------------------*/
void        dfilter()
{
/* Find the s-plane pole positions */
            temp = B1*B1 - 4*B0*B2;

            if(B2==0){                              /* a single pole */
                  p1=-B0/B1;
                  a0=A0/B1;
                  b1=-exp(p1*T);
            }
            if(temp>0){                             /* real and unequal poles */
                        pr=-B1/(2*B2);
                        pi=(pr*pr)-B0/B2;
                        pi=sqrt(pi);
                        p1=pr+pi;
                        p2=pr-pi;
                        c1=(A0+A1*p1)/((p1-p2)*B2);
                        c2=A1/B2-c1;
                        a0=c1+c2;
                        a1=-(c1*exp(p2*T) + c2*exp(p1*T));
                        b1=-exp(p1*T)-exp(p2*T);
                        b2=exp((p1+p2)*T);
            }
            if(temp<0){                             /* complex conjugate poles */
                        pr=-B1/(2*B2);
                        pi=(pr*pr)-B0/B2;
                        pi=sqrt(-pi);
                        cr=A1/(B2*2);
                        ci=-(A0+A1*pr)/(2*pi*B2);
                        a0=2*cr;
                        a1=-(cr*cos(pi*T)+ci*sin(pi*T))*2*exp(pr*T);
                        b1=-2*exp(pr*T)*cos(pi*T);
                        b2=exp(2*pr*T);
            }
            printf("discrete filter coeffs: \n");
            printf("a0 a1 a2: \t%f %f %f \n", a0, a1, a2);
            printf("b0 b1 b2: \t%f %f %f \n", b0, b1, b2);
}
```

给定模拟传递函数 $H(s)$的系数值后，程序 8A.1 里的 C 程序计算等价 z 平面传递函数 $H(z)$的系数。为了了解这个程序是如何工作的，我们先建立 $H(s)$和 $H(z)$的系数间的关系。

利用部分分式展开，8A.1 式的 s 平面传递函数可以表示为

$$\frac{B_0/A_2+(B_1/A_2)s}{A_0/A_2+(A_1/A_2)s+s^2}=\frac{c_1}{s-p_1}+\frac{c_2}{s-p_2} \tag{8A.3}$$

其中 p_1 和 p_2 是 $H(s)$的 s 平面的极点，如下式给定：

$$p_{1,2}=\frac{-A_1}{2A_2}\pm\left[\left(\frac{A_1}{2A_2}\right)^2-\frac{A_0}{A_2}\right]^{1/2} \tag{8A.4}$$

在 8A.3 式的两边同时乘上$(s-p_1)(s-p_2)$，并使 s 系数项和常数项相等，我们有

$$\frac{B_0}{A_2}=-(c_1p_2+c_2p_1) \tag{8A.5a}$$

$$\frac{B_1}{A_2}=c_1+c_2 \tag{8A.5b}$$

解出 c_1 和 c_2，我们有

$$c_1=\frac{B_0+B_1p_1}{(p_1-p_2)A_2} \tag{8A.6a}$$

$$c_2=\frac{B_1}{A_2}-c_1 \tag{8A.6b}$$

对 8A.3 式应用冲激不变转换，得出离散传递函数 $H(z)$：

$$\begin{aligned}H(z)&=\frac{c_1+c_2-(c_1\mathrm{e}^{p_2T}+c_2\mathrm{e}^{p_1T})z^{-1}}{1-(\mathrm{e}^{p_1T}+\mathrm{e}^{p_2T})z^{-1}+\mathrm{e}^{(p_1+p_2)T}z^{-2}}\\&=\frac{b_0+b_1z^{-1}}{1+a_1z^{-1}+a_2z^{-2}}\end{aligned} \tag{8A.7}$$

其中

$$b_0=c_1+c_2,\qquad b_1=-(c_1\mathrm{e}^{p_2T}+c_2\mathrm{e}^{p_1T})$$

$$a_1=-(\mathrm{e}^{p_1T}+\mathrm{e}^{p_2T}),\qquad a_2=\mathrm{e}^{(p_1+p_2)T}$$

p_1 和 p_2 在 8A.4 式中定义，c_1 和 c_2 在 8A.6 式中定义。

因此，给定二阶滤波器的 s 平面系数（即 A_0、A_1、A_2、B_0 和 B_1），利用上面的关系式可以直接得到等价的离散滤波器的系数。8A.7 式中 $H(z)$系数的计算，依赖于 s 平面极点 p_1 和 p_2 的类型。在实际应用中，有三种情况出现：当两个极点是(i)实数且不相等，(ii)复共轭对，(iii)实数且相等（也就是重合的极点）。仅考虑前两种情况，因为第 3 种很少发生且比较麻烦。

在第一种情况下，可以直接利用 8A.7 式求 $H(z)$的系数。在第二种情况下，利用 8A.7 式的更简单的形式避免复数算法。利用极点的特性，8A.7 式（对第二种情况）变为

$$\begin{aligned}H(z)&=\frac{(c_1+c_1^*)-(c_1\mathrm{e}^{p_1^*T}+c_1^*\mathrm{e}^{p_1T})z^{-1}}{1-(\mathrm{e}^{p_1T}+\mathrm{e}^{p_1^*T})z^{-1}+\mathrm{e}^{(p_1+p_1^*T)}z^{-2}}\\&=\frac{2c_\mathrm{r}-[c_\mathrm{r}\cos(p_\mathrm{i}T)+c_\mathrm{i}\sin(p_\mathrm{i}T)]2\mathrm{e}^{p_\mathrm{r}T}z^{-1}}{1-2\mathrm{e}^{p_\mathrm{r}T}\cos(p_\mathrm{i}T)z^{-1}+\mathrm{e}^{2p_\mathrm{r}T}z^{-2}}\end{aligned} \tag{8A.8}$$

其中 p_r 是 p_1 的实部，p_i 是 p_1 的虚部，c_r 是 c_1 的实部，c_i 是 c_1 的虚部。从 8A.4 式可得，p_1 的实部和虚部为

$$p_r = -\frac{A_1}{2A_2}, \quad p_i = \left\{-\left[\left(\frac{A_1}{2A_2}\right)^2 - \frac{A_0}{A_2}\right]\right\}^{1/2}$$

由 8A.6 式，部分因式系数 c_1 由如下给定：

$$c_1 = \frac{B_1}{2A_2} - \frac{B_0 + B_1 p_r}{2p_i A_2}\mathrm{j} = c_r + c_i\mathrm{j}$$

因此，对于 $H(s)$的极点是复数共轭的情况，二阶 z 传递函数的系数的标准形式是

$$b_0 = 2c_r, \qquad b_1 = -[c_r\cos(p_i T) + c_i\sin(p_i T)]2\mathrm{e}^{p_r T}$$

$$a_1 = -2\mathrm{e}^{p_r T}\cos(p_i T), \qquad a_2 = \mathrm{e}^{2p_r T}$$

例 8A.1　我们利用程序 8A.1 里的 C 语言程序来计算例 8.4 里的离散滤波器的系数。

程序期望频率被归一化。使用 1280 Hz 的抽样频率和 150 Hz 的截止频率，归一化的截止频率是 150/1280。传递函数首先通过用 s/α 来代替 s 来对频率进行伸缩变换，其中 $\alpha = 2\pi \times 150/1280 = 0.736\,31$：

$$H'(s) = \frac{\alpha^2}{\alpha^2 + \sqrt{2}\alpha s + s^2} = \frac{1}{1 + (\sqrt{2}/\alpha)s + (1/\alpha^2)s^2} = \frac{1}{1 + 1.920\,675s + 1.844\,96s^2}$$

程序的提示和输出如下所示。离散系数和例 8.4 里得到的相同。

```
impulse invariant discrete filters
enter s-plane coefficients
enter denominator coeffs: A0, A1, A2
1   1.920675   1.84496
enter numerator coeffs: B0, B1
1   0
discrete filter coeffs:
b0  b1  b2:  0.000000  0.307718  0.000000
a0  a1  a2:  1.000000  -1.030953  0.353088
```

由上面的列表，z 变换函数可以直接写为

$$H(z) = \frac{0.307\,718z^{-1}}{1 - 1.030\,953z^{-1} + 0.353\,088z^{-2}}$$

系数和例 8.4 里得到的值相同。

8B　用 MATLAB 设计 IIR 滤波器

MATLAB 信号处理工具箱，在给定一系列性能规范的情况下（例如通带和阻带频率，通带波纹和阻带衰减），对经典数字 IIR 滤波器（例如巴特沃斯，切比雪夫类型 I 和 II，以及椭圆滤波器）的设计和分析提供了很多有用的函数。

特别是，正如正文中所讨论的那样，工具箱对于把经典模拟滤波器转换成等价离散时间滤波器提供了相应的函数。

读者回顾前面可知，在数字 IIR 滤波器的设计里，一个关键的步骤就是系数计算。对于经典数字 IIR 滤波器，本阶段包含的步骤可以如下总结：

(1) 为希望的滤波器指定性能规范；

(2) 确定一个合适的具有巴特沃斯、切比雪夫类型 I、切比雪夫类型 II 以及椭圆特性的模拟原型低通滤波器；

(3) 转换这个原型模拟滤波器，生成一个低通、高通、带通或带阻滤波器；

(4) 把这个转换的滤波器再转换到等价的离散时间滤波器（例如，利用冲激不变或者双线性 z 变换法）。

MATLAB 信号处理工具箱提供了许多同时或分别实现步骤 2 到 4 的高级函数。例如，要建立一个具有巴特沃斯特性的低通、高通、带通或者带阻滤波器的 MATLAB 命令的语法是（步骤 1 和 2）：

```
[b, a] = butter (N, Wc, options)
[z, p, k] = butter (N, Wc, options)
```

第一个命令是计算 N 阶离散时间的巴特沃斯滤波器的分子和分母的系数，这个滤波器的 3 dB 截止频率 W_c（或者带沿频率）用奈奎斯特频率进行归一化。滤波器的分子和分母系数分别以矢量 b 和 a 输出，是以 z 的负幂上升的形式。

如果单词 "options" 被省略，那么命令默认为是对低通滤波器的（除非 W_c 是一个频率矢量，这种情况下它默认为是对带通滤波器的）。对于高通和带阻滤波器，将单词 "high" 和 "stop" 用做选项。对于带通和带阻滤波器，W_c 是一个两元素矢量，它定义了截止（或者带沿）频率：

$$W_c = [\omega_{c1}, \omega_{c2}]$$

其中 $\omega_{c1} < \omega < \omega_{c2}$ 是通带（带通滤波器）或者阻带（带阻滤波器）。

第二个命令返回零点和极点的位置，它是用 z、p 以及滤波器增益 k 来表示的矩阵。

对于其他经典滤波器类型也存在相似的命令。例如，对于切比雪夫类型 I 和 II，以及椭圆滤波器，MATLAB 命令的句法是

```
[b, a] = cheby1 (N, Ap, Wc, options)
[z, p, k] = cheby1 (N, Ap, Wc, options)

[b, a] = cheby2 (N, As, Wc, options)
[z, p, k] = cheby2 (N, As, Wc, options)

[b, a] = ellip (N, Ap, As, Wc, options)
[z, p, k] = ellip (N, Ap, As, Wc, options)
```

其中 Ap 和 As 分别是用 dB 表示的通带波纹以及阻带衰减。

有许多其他的有用命令可以用于在系数计算过程中执行中间过渡任务。例如，buttord、cheby1ord、ellipord 命令可以用于确定合适的滤波器的阶数。利用 butterp、cheby1p、cheby2p 和 ellipp 命令，可以确定合适的原型模拟低通滤波器的参数。

例 8B.1　利用冲激不变法和 MATLAB 来设计简单的低通滤波器　一个具有巴特沃斯特性的低通 IIR 滤波器，要求满足下面的性能规范：

截止频率	150 Hz
抽样频率	1.28 kHz
滤波器阶数，N	2

(a) 利用冲激不变法和 MATLAB，

(i) 求频率伸缩的模拟滤波器的系数、极点和零点；

(ii) 求 IIR 离散滤波器的系数、极点和零点，写出它的传递函数。

(b) 画出离散滤波器的幅度－频率响应和极零图。

解：

本题的 MATLAB m 文件在程序 8B.1 中给出。

(a) (i) 利用 m 文件，模拟滤波器的系数、极点和零点是

Coefficients:	b = 1.0e+005*[0, 0, 8.8826]
	a = 1.0e+005*[0.00001, 0.0133, 8.8826]
Poles:	1.0e+002*(-6.6643 ± 6.6643j)
Zeros:	None
Gain	8.8826e+005

程序 8B.1　例 8B.1 的 MATLAB m 文件

```
%
%  Program name: EX8B1.m
%  A simple lowpass filter
%
N=2;                                  %  Filter order
Fs=1280;                              %  Sampling frequency
fc=150;                               %  Cutoff frequency
WC=2*pi*fc;                           %  Cutoff frequency in radians
[b, a]=butter (N,WC,'s');             %  Create the analog filter
[z, p, k]=butter (N, WC, 's');
[bz, az]=impinvar (b, a, Fs);         %  Convert into discrete filter
subplot (2,1,1)                       %  Plot magnitude freq. response
[H, f]=freqz(bz, az, 512, Fs);
plot(f, 20*log10(abs(H)))
xlabel('Frequency (Hz)')
ylabel('Magnitude Response (dB)')
subplot(2,1,2)                        %  Plot pole-zero diagram
zplane(bz, az)
zz=roots(bz);                         %  Determine poles and zeros
pz=roots(az);
```

(ii) 利用 m 文件，离散时间滤波器的系数、极点和零点是

Coefficients:	b = [0 0.3078 0]
	a = [1.0000 −1.0308 0.3530]
Poles:	0.5154 ± 0.2955j
Zero:	0

利用这些系数，给出滤波器的传递函数为

$$H(z)=\frac{0.3078z^{-1}}{1-1.0308z^{-1}+0.3553z^{-2}}$$

这个结果和例 8.4 的相同。

(b) 图 8B.1 画出了幅度－频率响应和极零图。

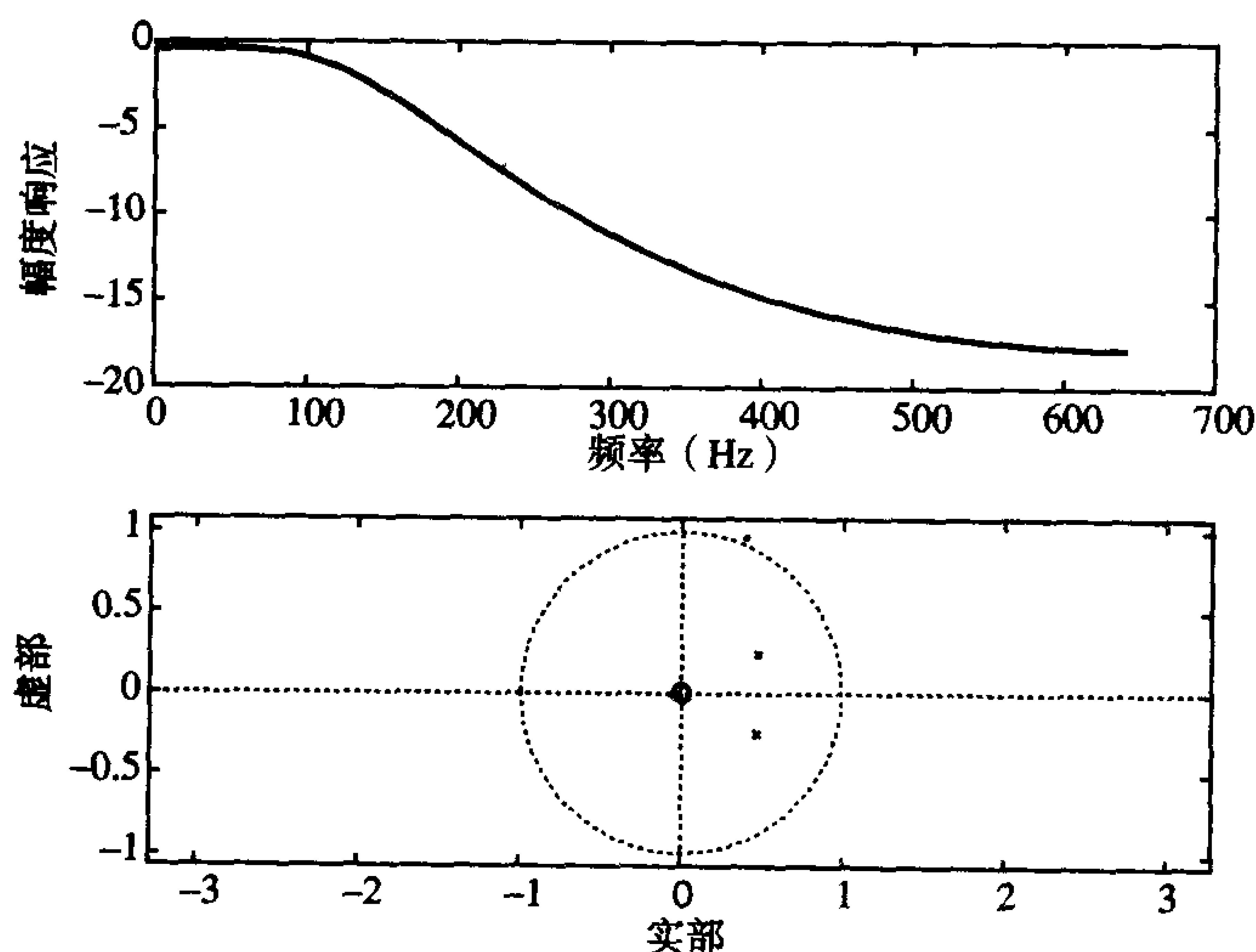


图 8B.1　幅度－频率响应和极零图

例 8B.2　利用双线性 z 变换法和 MATLAB 设计简单的低通滤波器　设计一个具有巴特沃斯特性的数字 IIR 滤波器，要求满足下面的性能规范：

截止频率	150 Hz
抽样频率	1.28 kHz
滤波器阶数	2

(a) 利用双线性 z 变换法和 MATLAB，确定离散滤波器的系数、极点和零点。

(b) 画出离散滤波器的幅度 – 频率响应和极零图。

解：

MATLAB m 文件在程序 8B.2 中列出。

(a) 在 m 文件的帮助下，求出 IIR 滤波器的系数矢量（b 和 a）、零点和极点（z 和 p）分别是

b = [0.0878, 0.1756, 0.0878]
a = [1.0000, −1.0048, 0.3561]
z = [−1, −1]
p = [0.5024 ± 0.3220j]
k = 0.0878

利用这些系数，滤波器的传递函数为

$$H(z) = \frac{0.3078z^{-1}}{1 - 1.0308z^{-1} + 0.3553z^{-2}}$$

(b) 在图 8B.2 里画出了幅度 – 频率响应和极零图。

程序 8B.2　例 8B.2 的 MATLAB m 文件

```
%
%  Program name: EX8B2.m
%  A simple lowpass filter
%
N=2;                                  %  Filter order
Fs=1280;                              %  Sampling frequency
FN=Fs/2;
fc=150;                               %  Cutoff frequency
Fc=fc/FN;                             %  Normalized Cutoff frequency
[b,a]=butter (N,Fc);                  %  Create and digitize analog filter.
[z,p,k]=butter (N, Fc);
subplot(2,1,1)                        %  Plot magnitude freq. response
[H, f]=freqz (b, a, 512, Fs);
plot(f, abs(H))
xlabel('Frequency (Hz)')
ylabel('Magnitude Response ')
subplot(2,1,2)                        %  Plot pole-zero diagram
zplane(b, a)
```

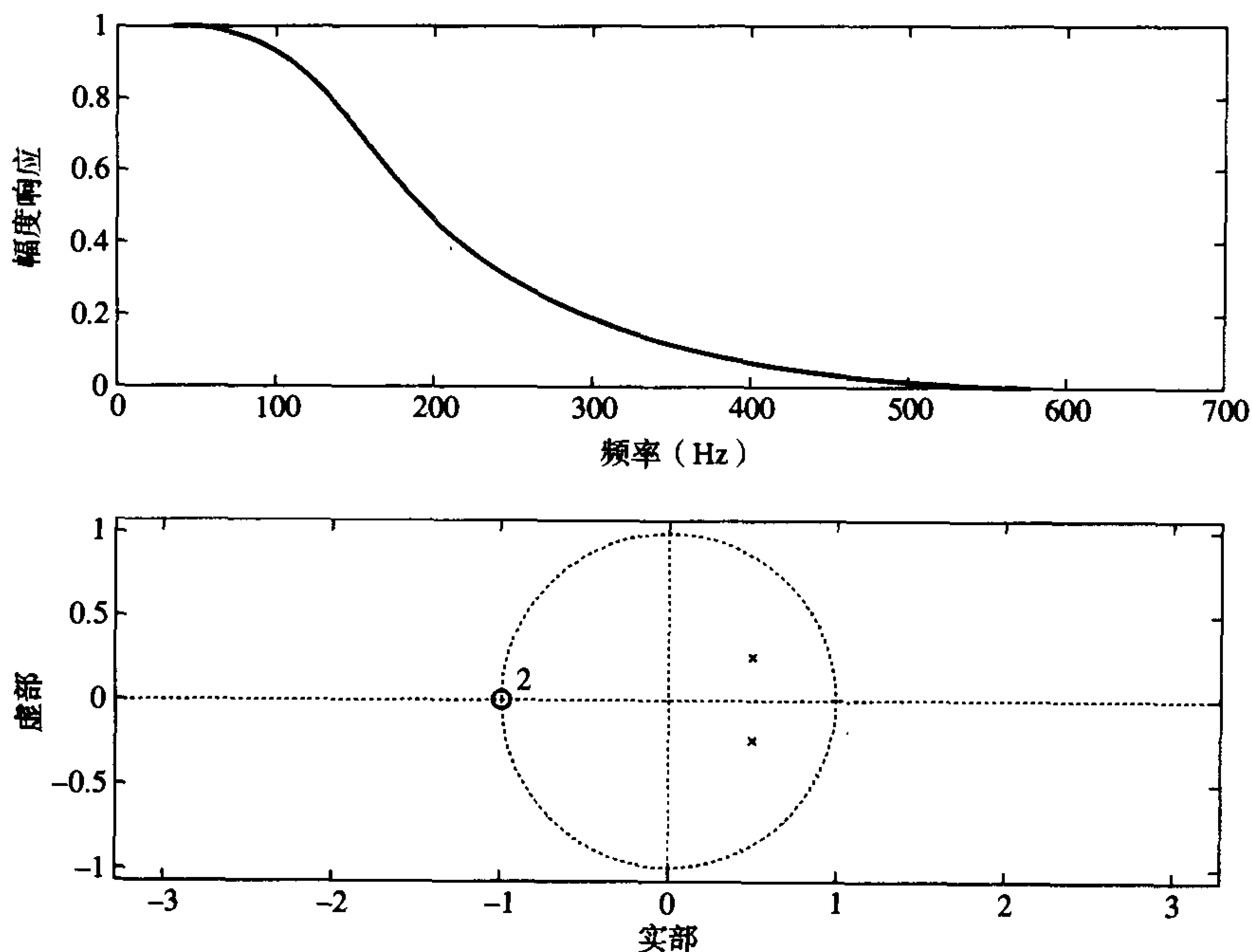


图 8B.2 幅度-频率响应和极零图

例 8B.3 利用双线性变换法和 MATLAB 设计简单的带通滤波器

(a) 计算具有巴特沃斯特性的离散带通滤波器的系数，要求满足下面的性能规范：

通带	200～300 Hz
抽样频率	2000 Hz
滤波器阶数	8

(b) 画出它的幅度-频率响应和极零图。

解：

MATLAB m 文件在程序 8B.3 中列出。

(a) 系数 b 和 a，零点、极点和增益 z、p 和 k 分别给出（注意，在 m 文件里利用的滤波器阶数在问题中已经指定；对于带通和带阻滤波器，阶数是 $2N$）：

```
b = [0.0004, 0, -0.0017, 0, 0.0025, 0, -0.0017, 0, 0.0004]
a = [1.0000, -5.1408, 13.1256, -20.9376, 22.6962, 17.0342, 8.6867, 2.7672, 0.4383]
z = [1, 1, 1, 1, -1, -1, -1, -1]
p = [0.5601 ± 0.7475j,  0.5800 ± 0.6286j, 0.6656 ± 0.5628j, 0.7647 ± 0.5648j]
k = 4.1660e-004
```

(b) 在图 8B.3 里画出了滤波器的幅度-频率响应以及极零图。

程序 8B.3 例 8B.3 的 MATLAB m 文件

```
%
% Program name: EX8B3.m
% A simple bandpass filter
%
Fs=2000;                          % Sampling frequency
FN=Fs/2;
fc1=200/FN;
fc2=300/FN;
[b,a]=butter (4,[fc1, fc2]);      % Create/digitize analog filter.
```

```
[z,p,k]=butter (4, [fc1, fc2]);
subplot(2,1,1)                              % Plot magnitude freq. response
[H, f]=freqz (b, a, 512, Fs);
plot(f, abs(H))
xlabel ('Frequency (Hz)')
ylabel ('Magnitude Response ')
subplot (2,1,2)                             % Plot pole-zero diagram
zplane (b, a)
```

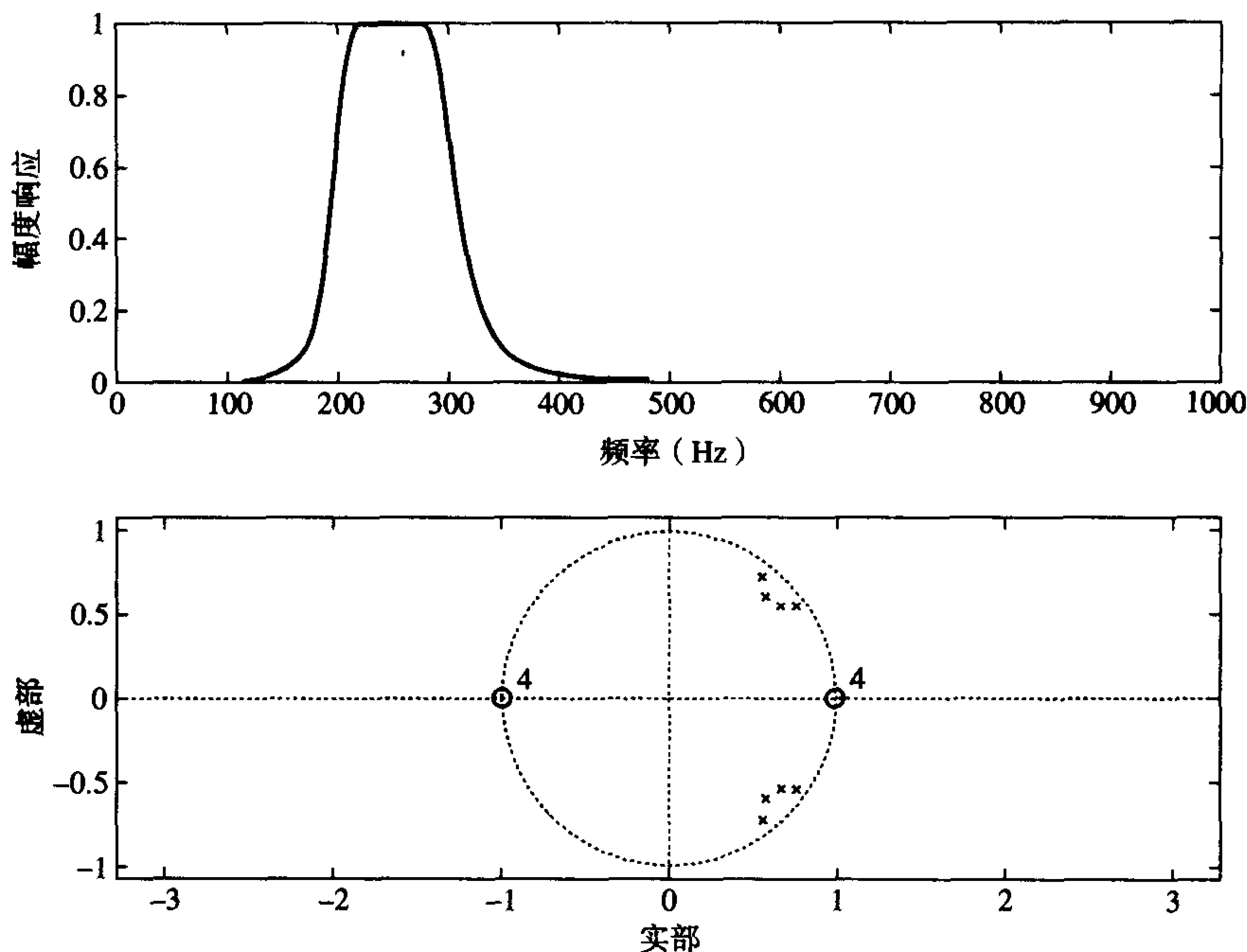


图 8B.3　幅度频率响应和极零图

例 8B.4　在指定通带和阻带边沿频率以及通带波纹 / 阻带衰减的情况下设计低通滤波器

设计一个具有巴特沃斯特性的数字低通滤波器，要求满足下面的性能规范：

通带	0 ~ 500 Hz
阻带	2 ~ 4 kHz
通带波纹	3 dB
阻带衰减	20 dB
抽样频率	8 kHz

(a) 利用双线性 z 变换和 MATLAB，求离散时间滤波器的阶数 N 和系数。

(b) 画出滤波器的幅度 - 频率响应和极零图。

解：

这个例子的第一部分类似于例 8.11，例 8.11 是通过手工计算解出的。这里我们利用 MATLAB。可以利用 MATLAB 命令来确定一个模拟滤波器的阶数（buttord 命令），利用 buttap 命令来确定原型模拟低通滤波器的极点和零点，利用 butter 命令来确定中间过程的模拟滤波器的参数，以及利用 bilinear 命令来确定离散时间滤波器的系数。然而，我们这里将利用 butter 命令的数字域形式，因为可以自动执行中间步骤来求滤波器系数。

(a) 在程序 8B.4 例给出了这个问题的 MATLAB 实现的 m 文件。

滤波器阶数N、零点和极点（zz和pz）、增益（kz）以及滤波器的系数（b和a）是

```
N  = 2
zz = [-1, -1]
pz = [0.7271 ± 0.2130j]
kz = 0.0300
b  = [0.0300    0.0599    0.0300]
a  = [1.0000   -1.4542    0.5741]
```

这些系数和例 8.11 里得到的系数是相同的，传递函数是

$$H(z) = \frac{0.030(1 + 2z^{-1} + z^{-2})}{1 - 1.4542z^{-1} + 0.5741z^{-2}}$$

(b) 图 8B.4 里画出了幅度 - 频率响应和极零图。

程序 8B.4　例 8B.4 的 MATLAB m 文件

```
%
%  Program name: EX8B4.m
%  Lowpass filter
%
Fs=8000;                                    %  Sampling frequency
Ap=3;
As=20;
wp=500/4000;
ws=2000/4000;
[N, wc]=buttord(wp, ws, Ap, As);            %  Determine filter order
[zz, pz, kz]=butter (N,500/4000);           %  Digitize filter
[b, a]=butter(N, 500/4000);
subplot (2,1,1)                             %  Plot magnitude freq. response
[H, f]=freqz (b, a, 512, Fs);
plot(f, abs(H))
xlabel('Frequency (Hz)')
ylabel('Magnitude Response ')
subplot(2,1,2)                              %  Plot pole-zero diagram
zplane(b, a)
```

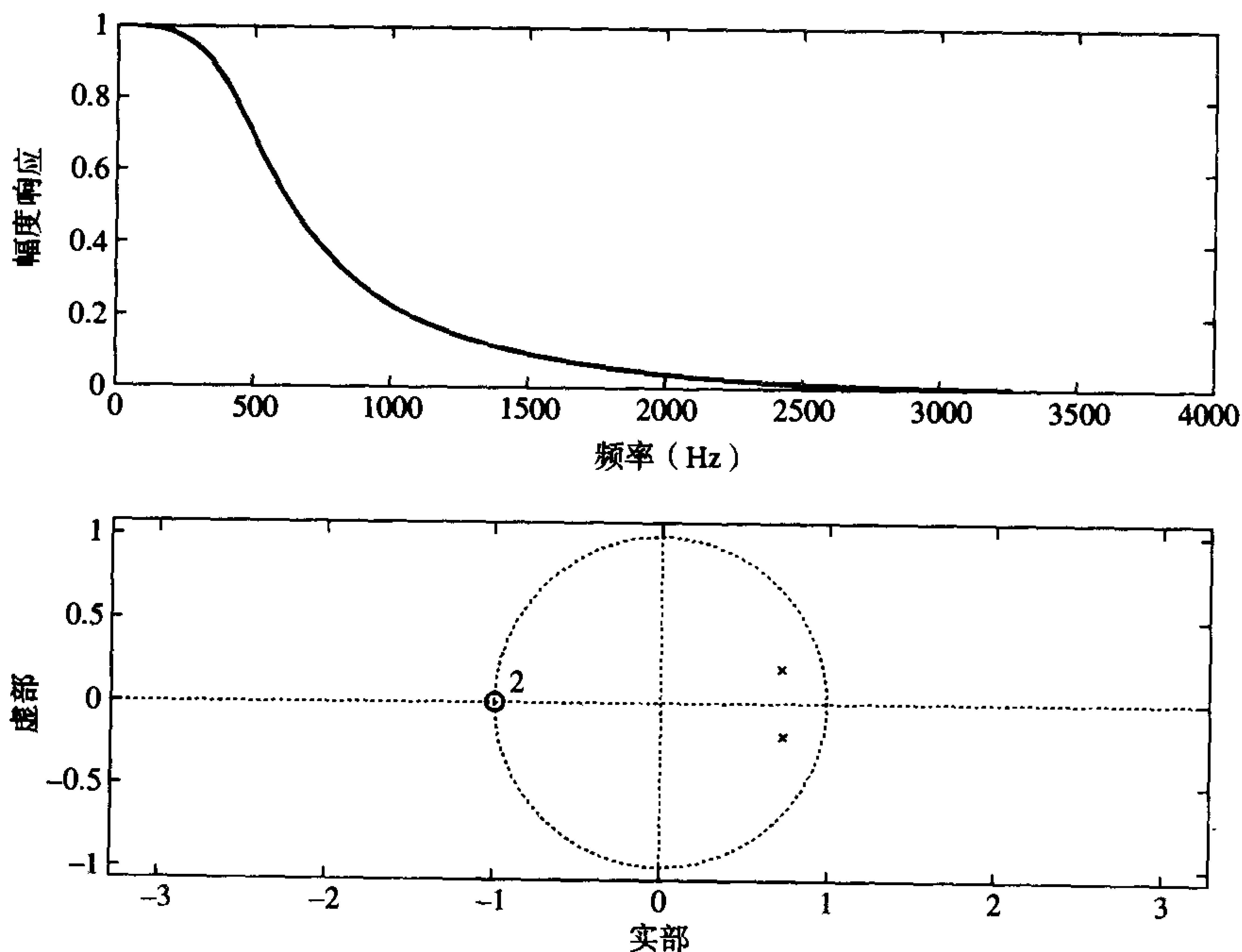


图 8B.4　幅度 - 频率响应和极零图

例 8B.5　在规定通带边沿频率、阻带边沿频率和通带波纹/阻带衰减的情况下设计一个高通滤波器　要求设计一个满足下面的性能规范的高通数字滤波器：

通带	2 ~ 4 kHz
阻带	0 ~ 500 Hz
通带波纹	3 dB
阻带衰减	20 dB
抽样频率	8 kHz

(a) 求合适的模拟原型低通滤波器的通带边沿频率和阻带边沿频率。

(b) 求原型低通滤波器的阶数 N。

(c) 利用双线性 z 变换求离散时间滤波器的系数和传递函数。

解：

(a) 这个问题的 MATLAB 实现的 m 文件在程序 8B.5 里给出。

(b) 滤波器阶数 N、极点和零点（zz 和 pz）、增益（kz）、滤波器的系数（b 和 a）是

```
N  = 2
zz = [1,1]
pz = ±0.4142j
kz = 0.2929
b  = [0.2929, -0.5858, 0.2929]
a  = [1.0000, 0.0000, 0.1716]
```

(c) 利用给出的系数，传递函数为

$$H(z) = \frac{0.2929(1 - 2z^{-1} + z^{-2})}{1 + 0.1716z^{-2}}$$

图 8B.5 给出了幅度 - 频率响应和极零图。

程序 8B.5　例 8B.5 的 MATLAB m 文件

```
%
% Program name: EX8B5.m
%
Fs=8000;                                  % Sampling frequency
Ap=3;
As=20;
wp=2000/4000;
ws=500/4000;
[N, wc]=buttord (wp, ws, Ap, As);         % Determine filter order
[zz, pz, kz]=butter (N, 2000/4000, 'high');   % Digitize filter
[b, a]=butter (N, 2000/4000, 'high');
subplot(2,1,1)                            % Plot magnitude freq. response
[H, f]=freqz (b, a, 512, Fs);
plot(f, abs(H))
xlabel('Frequency (Hz)')
ylabel('Magnitude Response ')
subplot(2,1,2)                            % Plot pole-zero diagram
zplane(b, a)
```

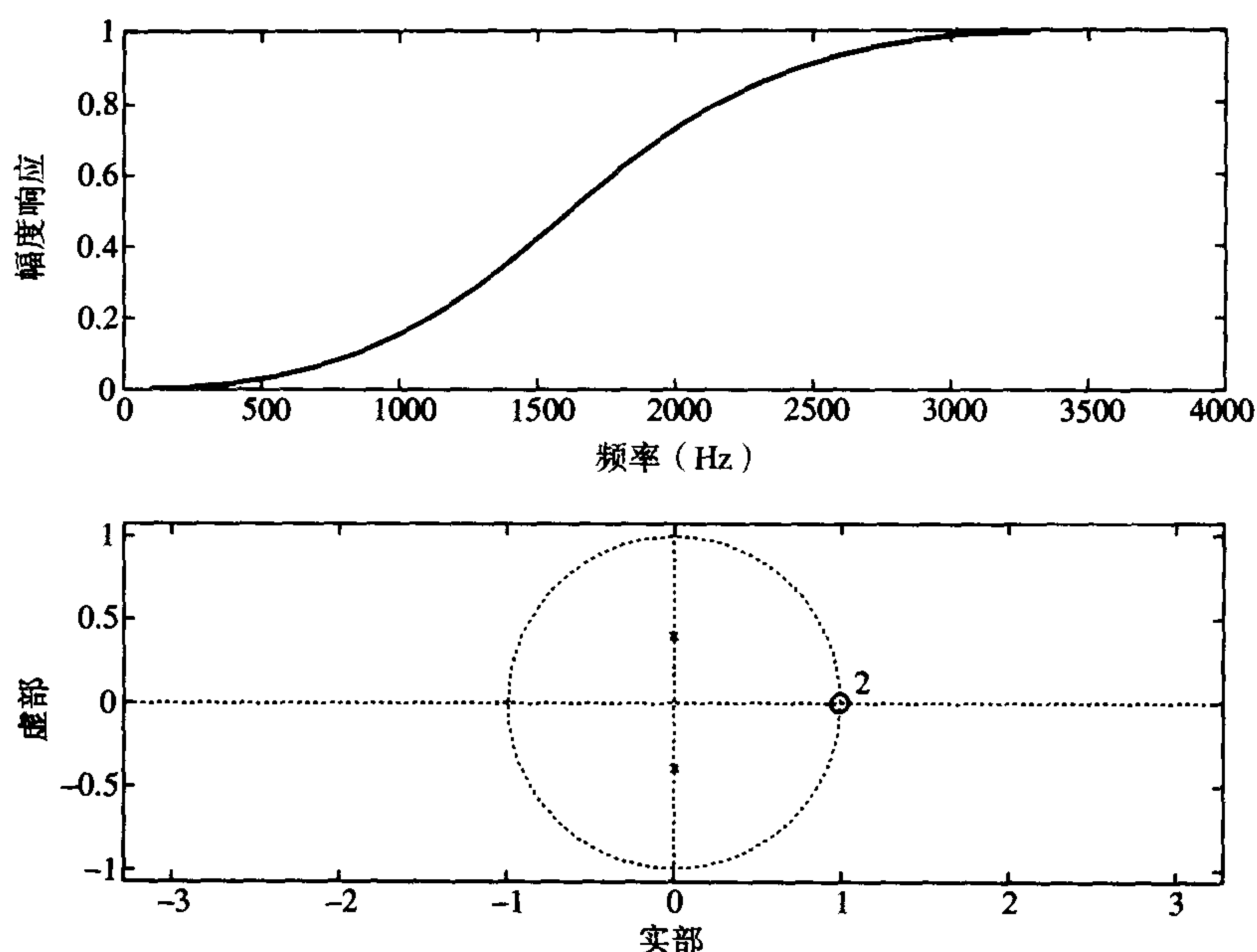


图 8B.5 幅度 – 频率响应和极零图

例 8B.6 在规定边沿频率和通带、阻带波纹的情况下设计带通滤波器 设计一个具有巴特沃斯幅度 – 频率响应的带通数字滤波器，它满足下面的性能规范：

通带下沿频率	200 Hz
通带上沿频率	300 Hz
阻带下沿频率	50 Hz
阻带上沿频率	450 Hz
通带波纹	3 dB
阻带衰减	20 dB
抽样频率	1 kHz

(a) 利 BZT 法和 MATLAB，

(i) 求滤波器阶数 N；

(ii) 求离散时间滤波器的极点、零点、增益和传递函数。

(b) 画出滤波器的幅度 – 频率响应和极零图。

解：

(a) 在程序 8B.6 里给出了 MATLAB 的 m 文件。利用 m 文件，得到滤波器的阶数 N，以及离散滤波器的极点和零点（zz 和 pz）、增益（kz）和系数（b 和 a）：

```
N  = 2 (the order of the bandpass filter is 2 * N, i.e. 4)
zz = [1, 1, -1, -1]
pz = [-0.1884 ± 0.7791j, -0.1884 ± 0.7791j]
kz = 0.0675
b  = [0.0675, 0, -0.1349, 0, 0.0675]
a  = [1.0000, -0.0000, 1.1430, -0.0000, 0.4128]
```

对应的传递函数 $H(z)$ 是

$$H(z)=\frac{0.0675(1-2z^{-2}+z^{-4})}{1+1.143z^{-2}+0.4128z^{-4}}$$

(b) 如果我们令 $N = 1$（就像在例 8.11 里那样），零点（zz）、极点（pz）、增益（kz）和系数（b 和 a）变为

```
zz = [-1, 1]
pz = [0.0000 ± 0.7138j]
kz = 0.2452

b  = [0.2452, 0, -0.2452]
a  = [1.0000, -0.0000, 0.5095]
```

对应的传递函数 $H(z)$ 是

$$H(z) = \frac{0.2452(1 - z^{-2})}{1 + 0.5095z^{-2}}$$

图 8B.6 给出了幅度 – 频率响应和极零图。

程序 8B.6　例 8B.6 的 MATLAB m 文件

```
%
%   Program name: EX8B6.m
%   Bandpass filter
%
Fs=1000;                                    %   Sampling frequency
Ap=3;
As=20;
Wp=[200/500, 300/500];                      %   Bandedge frequencies
Ws=[50/500, 450/500];
[N, Wc]=buttord (Wp, Ws, Ap, As);           %   Determine filter order
[zz, pz, kz]=butter (N,Wp);                 %   Digitize filter
[b, a]=butter (N, Wp);
subplot(2,1,1)                              %   Plot magnitude freq. response
[H, f]=freqz (b, a, 512, Fs);
plot(f, abs(H))
xlabel('Frequency (Hz)')
ylabel('Magnitude Response ')
subplot(2,1,2)                              %   Plot pole-zero diagram
zplane(b, a)
```

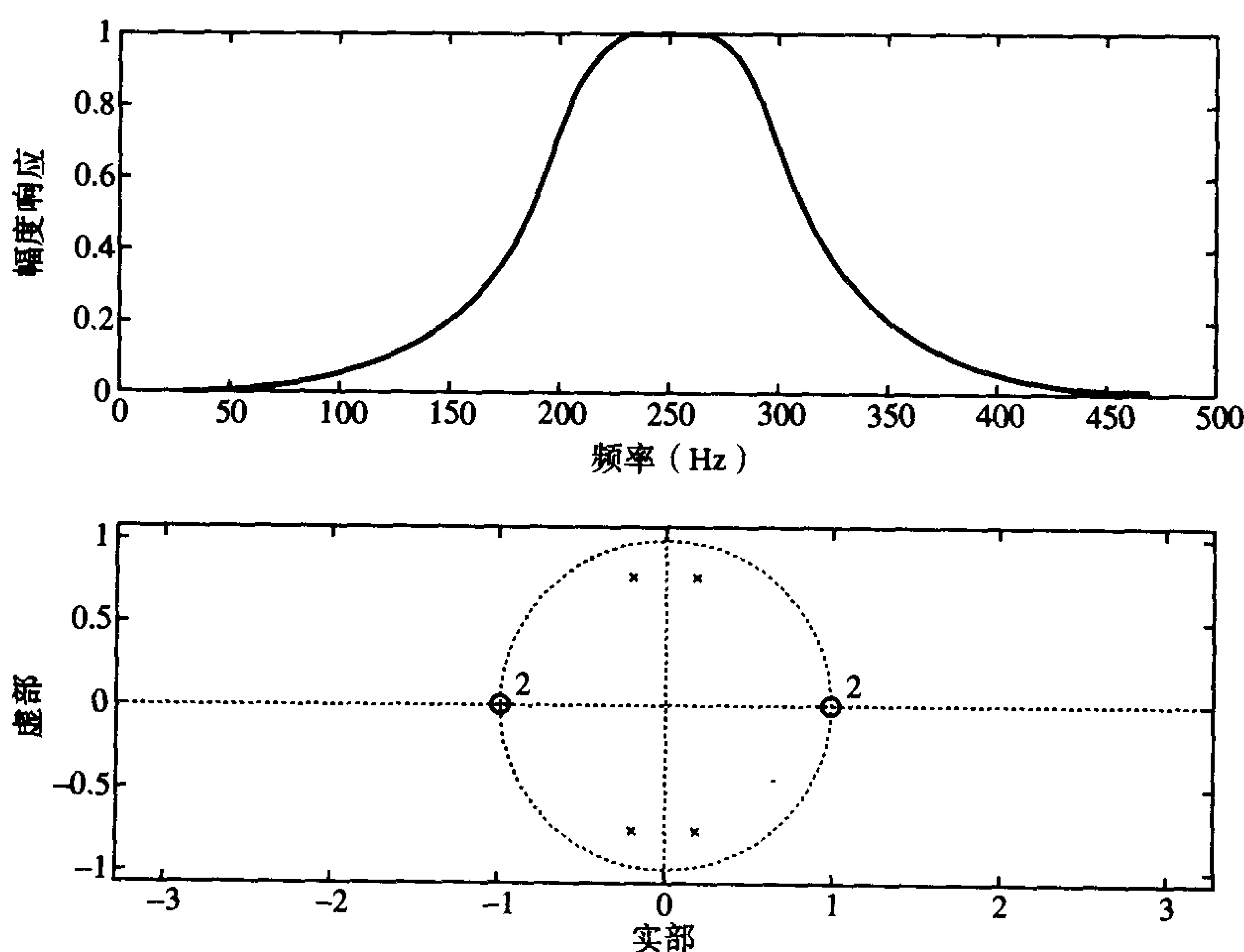


图 8B.6　幅度 – 频率响应和极零图

例 8B.7 在规定通带边沿频率、通带波纹阻带衰减情况下设计带阻滤波器 设计一个具有巴特沃斯幅度－频率响应的带阻数字 IIR 滤波器，它满足下面的性能规范：

下通带	0～50 Hz
上通带	450～500 Hz
阻带	200～300 Hz
通带波纹	3 dB
阻带衰减	20 dB
抽样频率	1 kHz

(a) 求合适的原型低通滤波器的通带和阻带边沿频率；

(b) 求原型低通滤波器的阶数 N；

(c) 利用 BZT 法求离散时间滤波器的系数和传递函数。

解：

在程序 8B.7 里给出了这个例子的 MATLAB m 文件。利用这个 m 文件，滤波器的阶数 N 以及极点和零点（zz 和 pz）、增益（kz）和系数是

```
N  = 2 (order of the bandstop filter = 2 * N, i.e. 4)
zz = two pairs of zeros at j and at -j
pz = [-0.1884 ± 0.7791j, 0.1884 ± 0.7791j]
kz = 0.6389
b = [0.6389, -0.0000, 1.2779, -0.0000, 0.6389]
a = [1.0000, -0.0000, 1.1430, -0.0000, 0.4128]
```

滤波器的传递函数是

$$H(z) = \frac{0.6389(1 + 2z^{-2} + z^{-4})}{1 + 1.143z^{-2} + 0.4128z^{-4}}$$

如果我们令 $N = 1$（如例 8.12 一样），传递函数变为

$$H(z) = \frac{0.7548(1 + z^{-2})}{1 + 0.5095z^{-2}}$$

图 8B.7 里给出了幅度－频率响应和极零图。

程序 8B.7 例 8B.7 的 MATLAB m 文件

```
%
%  Program name: EX8B7.m
%  Bandstop filter
%
Fs=1000;                                   %  Sampling frequency
Ap=3;
As=20;
Wp=[50/500, 450/500];                      %  Bandedge frequencies
Ws=[200/500, 300/500];
[N, Wc]=buttord (Wp, Ws, Ap, As);          %  Determine filter order
[zz, pz, kz]=butter (N,Ws, 'stop');        %  Digitize filter
[b, a]=butter (N, Ws, 'stop');
subplot (2,1,1)                            %  Plot magnitude freq. response
[H, f]=freqz (b, a, 512, Fs);
plot (f, abs (H))
xlabel ('Frequency (Hz)')
ylabel ('Magnitude Response')
subplot (2,1,2)                            %  Plot pole-zero diagram
zplane (b, a)
```

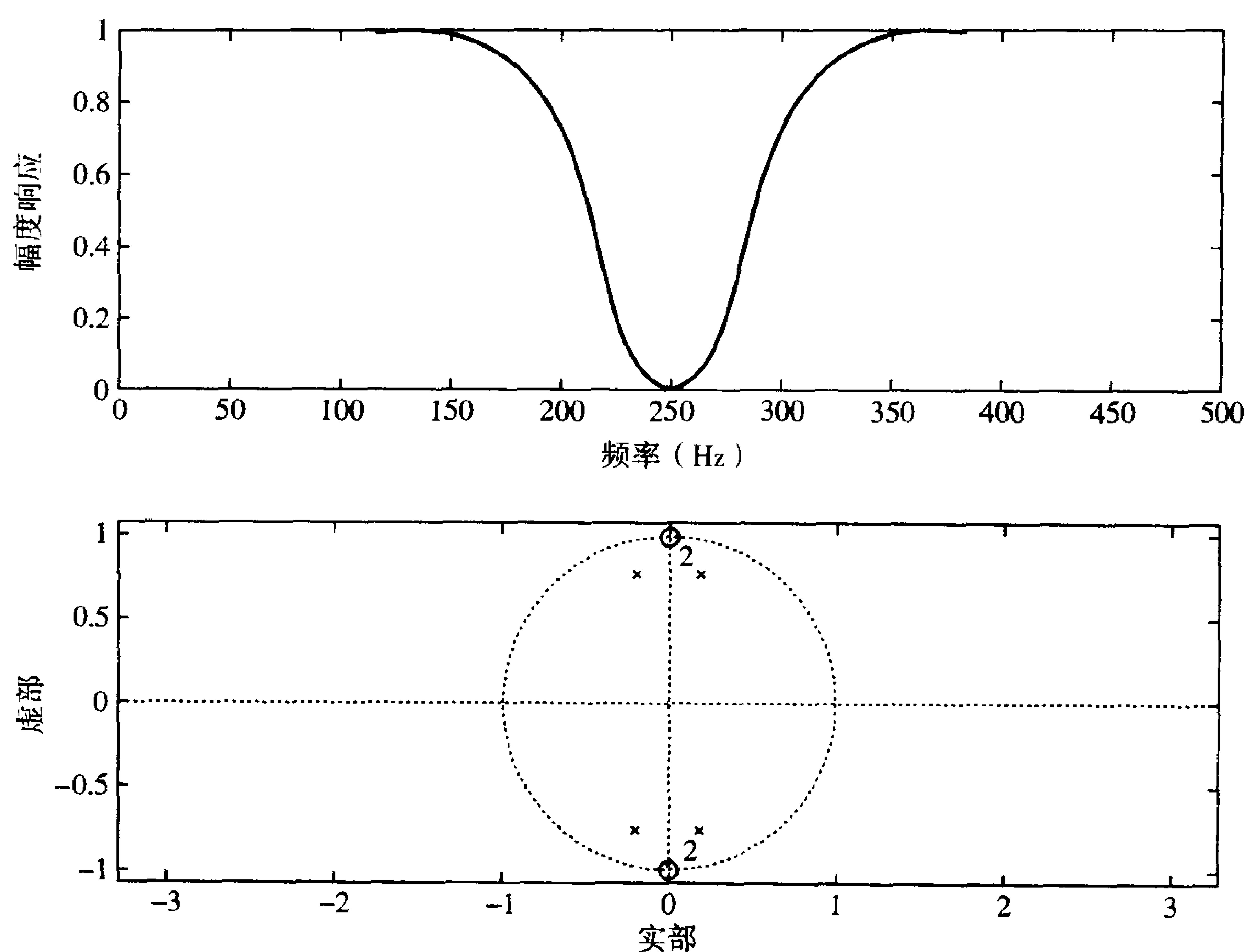


图 8B.7　幅度－频率响应和极零图

例 8B.8　椭圆数字带通滤波器的设计举例　设计一个满足下面频率响应规范的数字滤波器：

通带	20.5 ~ 23.5 kHz
阻带	0 ~ 19 kHz, 25 ~ 50 kHz
通带波纹	≤ 0.25 dB
阻带衰减	> 45 dB
抽样频率	100 kHz

(a) 利用 BZT 法和 MATLAB，求合适的滤波器的传递函数，该滤波器用适合于串联实现的二阶部分的形式表示。

(b) 画出这个滤波器的幅度－频率响应和极零图。

假设滤波器具有椭圆特性。

解：

(a) 在程序 8B.8 里列出了设计这个问题的 MATLAB m 文件。命令 ellipord 是用来确定一个合适的椭圆原型滤波器的阶数。命令 ellip 是用来确定滤波器的系数、传递函数的极点和零点。命令 zp2sos 用来把传递函数转换为二阶部分形式。二阶部分的系数是用矩阵 sos 返回的。

滤波器系数（b 和 a）、极点和零点（p 和 z）、增益（k）和二阶部分的矩阵 sos 在下面列出：

```
Filter order, N = 4
b = [0.0061, -0.0083, 0.0238, -0.0221, 0.0351, -0.0221, 0.0236, -0.0083, 0.0061]
a = [1.0000, -1.4483, 4.4832 -4.2207, 6.6475, -3.9458, 3.9187, -1.1828, 0.7634]
z = [-0.0118 ± 0.9999j, 0.3737 ± 0.9275j, -0.2553 ± 0.9669j, 0.5663 ± 0.8242j]
p = [0.0872 ± 0.9793j, 0.1352 ± 0.9404j, 0.2795 ± 0.9432j, 0.2223 ± 0.9246j]
k = 0.0061
sos =
     0.1203     0.0614     0.1203     1.0000     -0.2705     0.9026
```

```
0.2051   -0.2323   0.2051   1.0000   -0.4446   0.9042
0.3740    0.0088   0.3740   1.0000   -0.1744   0.9665
0.6642   -0.4965   0.6642   1.0000   -0.5589   0.9878
```

sos 是一个二阶部分的 4 × 6 矩阵，它有如下格式：

$$
\begin{matrix}
b_{01} & b_{11} & b_{21} & a_{01} & a_{11} & a_{21} \\
b_{02} & b_{12} & b_{22} & a_{02} & a_{12} & a_{22} \\
b_{03} & b_{13} & b_{23} & a_{03} & a_{13} & a_{23} \\
b_{04} & b_{14} & b_{24} & a_{04} & a_{14} & a_{24}
\end{matrix}
$$

很显然二阶部分的系数保存在每一行中，每一行包含一个二阶部分的系数。

在图 8B.8 里给出了幅度－频率响应图和极零图。

程序 8B.8　设计例 8B.8 的 MATLAB m 文件

```
%
% Program Name: EX8B.m
%
Ap=0.25;
As=45;
Fs=100000;
Wp=[20500/50000, 23500/50000];          % Bandedge frequencies
Ws=[19000/50000, 25000/50000];
[N,Wc]=ellipord(Wp, Ws, Ap, As);        % Determine filter order
[b, a]=ellip(N, Ap, As, Wc);            % Determine filter coeffs
[z, p, k]=ellip(N, Ap, As, Wc);         % Determine poles and zeros
sos=zp2sos (z, p, k);                   % Convert to second order sections
subplot(2,1,1)                          % Plot magnitude freq. response
[H, f]=freqz(b, a, 512, Fs);
plot(f, 20*log10 (abs(H)))
xlabel('Frequency (Hz)')
ylabel('Magnitude Response (dB)')
subplot(2,1,2)                          % Plot pole-zero diagram
zplane(b, a)
```

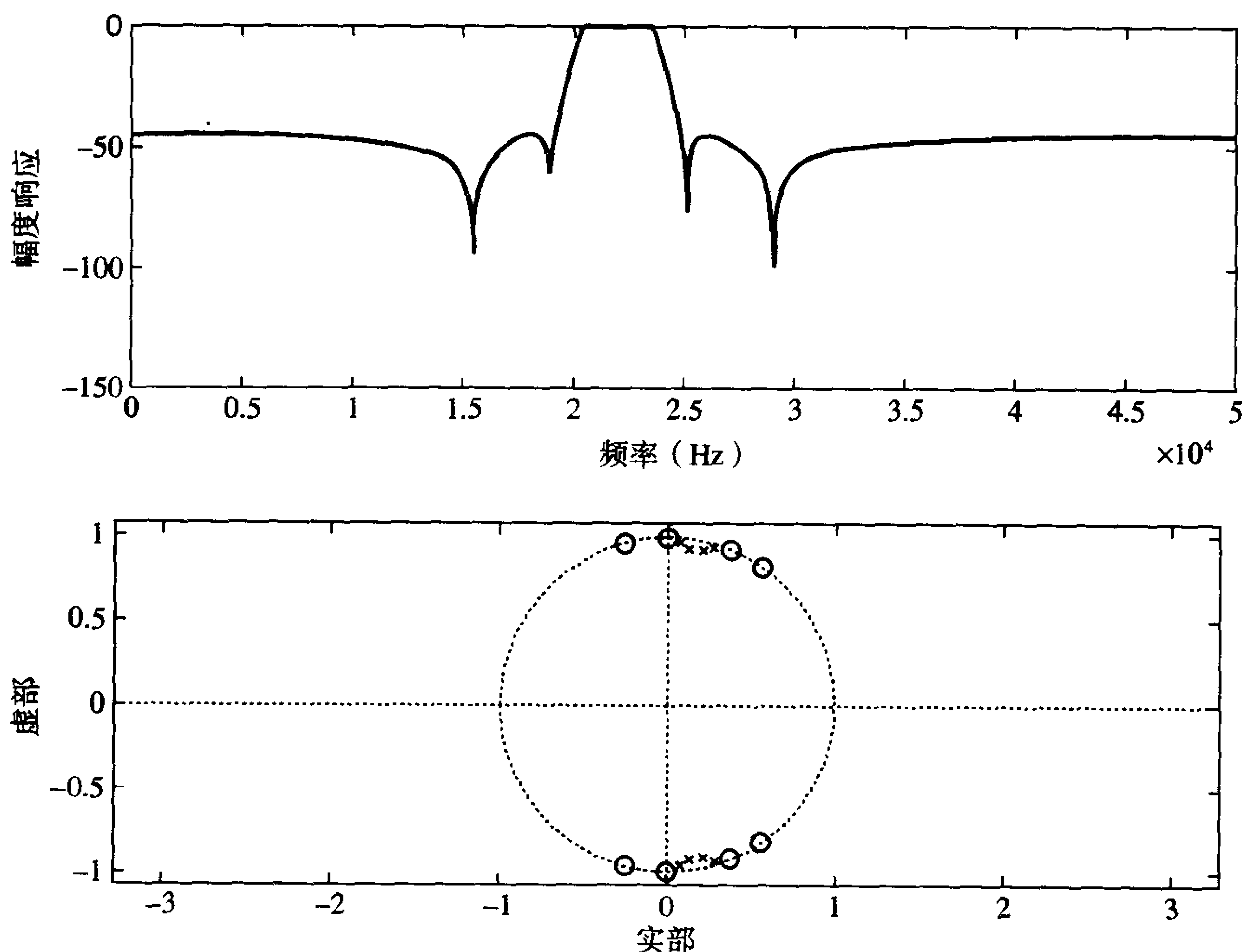


图 8B.8　幅度－频率响应和极零图

8C　利用实数算法来计算复平方根

给定用直角坐标系 $x+\mathrm{j}y$ 表示的一个复数，令它的平方根是 $u+\mathrm{j}v$：

$$u+\mathrm{j}v=(x+\mathrm{j}y)^{1/2} \tag{8C.1}$$

对 8C.1 式两边平方，我们有

$$u^2-v^2+\mathrm{j}2uv=x+\mathrm{j}y$$

令实部和实部相等、虚部和虚部相等，得

$$u^2-v^2=x \tag{8C.2a}$$

$$2uv=y \tag{8C.2b}$$

由 8C.2b 式，有

$$v=y/2u \tag{8C.2c}$$

把 8C.2c 式代入 8C.2a 式中，我们有

$$u^2-y^2/4u^2=x \tag{8C.3}$$

简化 8C.3 式后有

$$4u^4-4xu^2-y^2=0$$

这是 u^2 的二次型，它的解为

$$u^2=\frac{x}{2}+\frac{1}{8}(16x^2+16y^2)^{1/2}$$
$$=\frac{x+(x^2+y^2)^{1/2}}{2}$$

（注意负的解是不允许的，因为 $u^2\geqslant 0$。）解出 u 并利用 8C.2c 式，u 的解为

$$u=\left[\frac{x+(x^2+y^2)^{1/2}}{2}\right]^{1/2} \tag{8C.4a}$$

$$v=y/2u \tag{8C.4b}$$

或者

$$u=-\left[\frac{x+(x^2+y^2)^{1/2}}{2}\right]^{1/2} \tag{8C.5a}$$

$$v=y/2u \tag{8C.5b}$$

那么计算平方根 $x+\mathrm{j}y$ 的算法是

- 步骤 1：令 $u+\mathrm{j}v=(x+\mathrm{j}y)^{1/2}$
- 步骤 2：$u=\pm\left[\frac{x+(x^2+y^2)^{1/2}}{2}\right]^{1/2}=\pm\left(\frac{x+|x+\mathrm{j}y|}{2}\right)^{1/2}$
- 步骤 3：$v=\pm y/2u$

例如

$$(-1+\mathrm{j})^{1/2}=0.455\,09+1.098\,68\mathrm{j} \text{ and } -0.455\,09-1.098\,68\mathrm{j}$$
$$(-1-\mathrm{j})^{1/2}=0.455\,09-1.098\,68\mathrm{j} \text{ and } -0.455\,09+1.098\,68\mathrm{j}$$

第9章　多抽样率数字信号处理

本章从实用的观点出发，讨论了多抽样率处理涉及的主要内容。文中包含充足的工程实例以解释和阐述多抽样率处理的基本概念。同时还介绍了实际多抽样率系统的设计方法，使读者可以设计满足自己需要的处理系统。并提供一套C语言程序，用以在个人计算机上进行多抽样率处理的设计和软件实现。

9.1　引言

在现代数字系统中，随着对超过一种抽样率数据处理需求的日益增长，数字信号处理（DSP）领域的一个新分支——多抽样率处理（Crochiere and Rabiner, 1975, 1976, 1979, 1981, 1983, 1988）出现了。多抽样率处理中两个主要的操作是抽取和内插，它们使数据率能够方便地被改变。抽取降低了抽样率（或者说抽样频率），有效地压缩了数据，只保留所希望的信息。另一方面内插将增加抽样率。通常，改变数据率的目的是为了使处理更简便（例如计算效率更高），或者与其他系统相匹配。举个简单例子，如果将一个信号的抽样率从100 kHz压缩到只有10 kHz，且没有丢失所需的信息，那么在接下来的信号处理操作中，我们一下降低其负载到只有原先的十分之一。另一个例子，如果我们希望在一个具有48 kHz数据处理能力的数字音响上欣赏按44.1 kHz抽样率录制的光盘（CD）音乐，那么首先需要用多抽样率方法将光盘的数据速率增加到48 kHz。

本章及章末习题的目的在于对多抽样率信号处理的理论、操作和应用提供一个实用的理解。特定的学习目标有

(1) **方法/理论**　读者需要学习抽样率变换理论，特别是

- 降低抽样率的方法（抽取——抽样率降低和数字抗混叠滤波）；
- 增加抽样率的方法（内插——抽样率增加和数字去镜像滤波）；
- 多级抽样率变换的理论；
- 多相滤波的方法。

(2) **操作**　应能够根据指定的应用在结构图层次上设计多抽样率数据转换器，特别是

- 如何为多抽样率转换器指定、设计和分析相应的滤波器：
- 如何决定多抽样率转换器的参数；
- 如何评估多抽样率转换器的计算效率；
- 如何执行抽样率变换。

(3) **应用**　能够在音频工程、数字通信及生物医学等应用中正确使用多抽样率技术和方法，特别是

- 音频信号处理——过抽样（单比特）模/数转换（ADC）与数/模转换（DAC），CD唱机和数据获取；

- 数字通信——远程多路复用器，通信接收机；
- 生物医学——用于胎儿心电图（ECG）和脑电图（EEG）的窄带滤波器。

本章汲取了许多公开出版物的内容，特别是参考了为多抽样率处理做出巨大贡献的 Crochiere 和 Rabiner 的文章，在此我们深表感谢。

9.1.1 当前多抽样处理的一些工业应用

多抽样率处理具有很多优点，在现代数字系统中得到越来越多的应用。高质量数据获取及储存系统利用多抽样率技术来避免使用昂贵的抗混叠模拟滤波器，不同的抽样频率被有效地用于不同带宽的信号。这些应用的出发点是，如果一个模拟信号被一个远超过抽样定理需要的频率所抽样，则在其数字化前，可以使用一个很简单的抗混叠模拟滤波器。而在数字处理期间，信号可以使用多抽样处理方法而很容易地降低到希望的数据率。这种系统的一个好的例子是 EDR8000（Earth Data, UK）的磁带录音机。

在语音处理中，多抽样率技术用来降低语音数据的储存空间或传输率。语音参数估计的计算可以在用于数据存储或传输的较低抽样率上进行。如果需要，使用多抽样率技术将原始语音在高抽样率上从低比特率数据中重建。

在数字音响中使用便宜、高分辨率的模/数转换器（ADC）的需求导致在设计过程中过抽样技术的应用，取代了早先传统的逐次比较的调制编码技术（Adams, 1986; Agrawal and Shenoi, 1983; Claasen et al., 1980; Matsuya et al., 1987; Welland et al., 1989）。例如，通过过抽样，在模/数转换器中固有的量化噪声被分配到一个较宽的频带上，使带内噪声功率降到较低的电平，增加了模/数转换的有效位数。还有，由于其简洁性（例如，它不需要抽样-保持放大器）和低成本，可以使用 Σ-Δ 调制来设计这些高性能的模/数转换器。当今使用的大多数，即便不是全部的价廉的高分辨率模/数转换器(18、20、24位)都使用了多抽样率处理。这样的例子包括CS532X(Crystal Semiconductor)和 DSP56ADCx（摩托罗拉）。

多抽样率处理同样在有效利用DSP功能上具有重要的应用。例如，使用通常的DSP实现窄带数字FIR滤波器会遇到一系列的问题，这是由于此种滤波器需要非常多的参数来满足其陡直的频响要求。多抽样率技术提供了一个简便的解，允许滤波在一个较低的抽样速率上进行，极大地降低了滤波器阶数。多抽样率技术还能应用于许多其他场合，包括常用的CD播放器上。有关多抽样率技术的各种应用将在本章中陆续介绍。

9.2 多抽样率信号处理的概念

实现数字信号抽样率转换的一种最易理解的方法是将它先转变成模拟信号，再重新按新的抽样率数字化。数/模/数转换过程中的固有误差，如量化误差和混叠误差，会导致信号失真。由于信号已经数字化了，最好使用数字化处理直到必须转换到模拟信号，比如说到扬声器中。多抽样率处理基本上是一种有效改变数字信号抽样频率的方法，它的引人之处在于能够彻底发挥传统 DSP 的潜力。例如，在实时 DSP 系统中，抗混叠和去镜像滤波都可以在数字域进行，即同时达到了尖锐的幅频特性和线性相位响应。

抽取和内插处理是多抽样率信号处理的基本操作，它们允许信号抽样频率增加或降低而不带来显著的负面效应（如量化或混叠误差）。接下来我们给出这些操作的详细方法。

9.2.1　抽样率降低：按整数因子抽取

图9.1(a)给出了一个将信号$x(n)$按整数因子M抽取的框图。它包括一个数字抗混叠滤波器$h(k)$，以及一个抽样频率压缩器（或称抽取器），用一个向下的箭头及抽取因子M来表示。抽取器将抽样频率从F_s降低到F_s/M。为了防止输出低频信号谱混叠，在输入信号前面加入一个带限滤波器，使频率不超过$F_s/2M$。这样信号$x(n)$的带宽将被限制（有关抽取滤波器的指标将在9.3.1节详细讨论）。抽样频率降低是通过将滤波后信号$w(n)$每隔M个抽样并丢弃其中的$M-1$个来实现的。抽取过程的输入 - 输出关系式为

$$y(m) = w(mM) = \sum_{k=-\infty}^{\infty} h(k)x(mM-k) \tag{9.1a}$$

其中

$$w(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k) \tag{9.1b}$$

图9.1(b)阐明了对于$M=3$的简单抽取过程。这时，$x(n)$的每三个抽样中有两个被丢弃了。实质上，抽取是一个数据压缩操作。

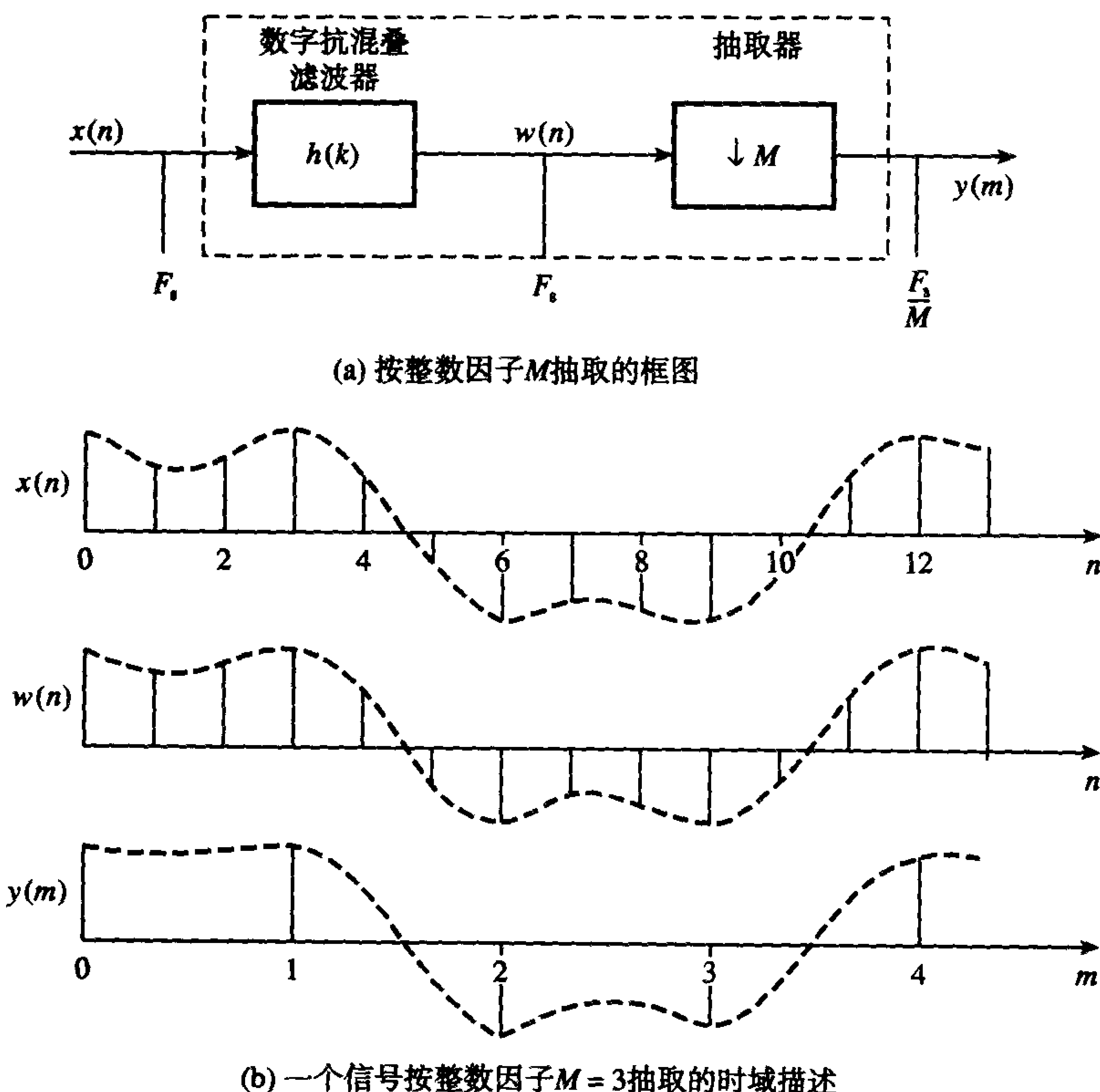


图9.1　按整数因子M的抽取过程，注意到$w(n)$每隔三个抽样只有一个输出

图9.2给出了抽取过程的频域描述，这里假定输入信号$x(n)$是一个宽带信号。图9.2(b)的虚线指示了输入信号频谱的镜像分量，如果没有在抽取前将其频带限制，它将导致信号$x(n)$频谱的混叠。

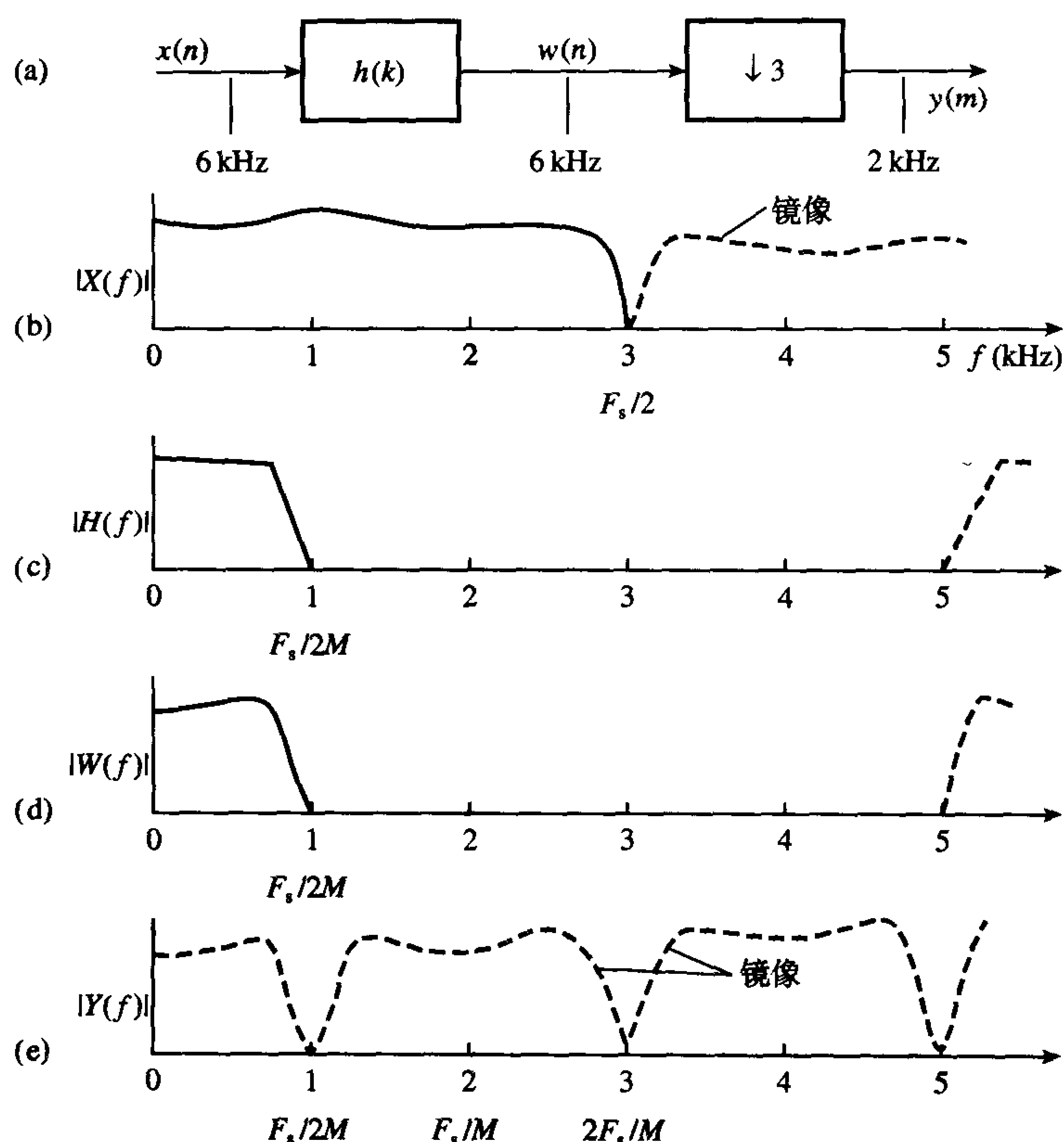


图9.2　一个信号从6 kHz到2 kHz抽取过程的频域描述。如果没有前置的数字滤波器，数字频谱的镜像分量可能混叠到信号中

9.2.2　抽样率增加：按整数因子内插

从许多方面来看，内插与数/模转换过程是数字等效的，因为数/模转换中模拟信号的重建是通过内插数字抽样点来实现的。然而，数字内插过程产生了一些特殊的数值。

假定信号$x(n)$的抽样频率为F_s，内插过程将抽样率增大L倍到LF_s。图9.3(a)给出抽取滤波器的框图。它包括一个抽样频率扩展器（或称内插器），用一个向上的箭头及内插因子M来表示，以及一个去镜像滤波器。对$x(n)$的每一个抽样，内插器在信号中增加L-1个值为零的抽样，这样输出信号$w(m)$的抽样频率变为LF_s。然后把该信号通过一个低通滤波器以去除抽样率改变带来的镜像频率，得到新信号$y(m)$。插入L-1个零值的结果是一个抽样的信号能量扩展到L个抽样，输出信号（平均）幅度被衰减到输入的L分之一。因此有必要通过将每个输出抽样乘以L来补偿这个衰减。

内插过程的输入-输出关系式为

$$y(m) = \sum_{k=-\infty}^{\infty} h(k)w(m-k) \tag{9.2a}$$

其中

$$w(m) = \begin{cases} x(m/L), & m = 0, \pm L, \pm 2L, \ldots \\ 0 \end{cases} \tag{9.2b}$$

图9.3(b)～图9.3(d)给出了$L=3$的简单情况下内插过程的时域描述。可以看到，对于$x(n)$的每个抽样，产生了3个输出抽样，这是由于内插器将两个零值抽样插入的缘故。

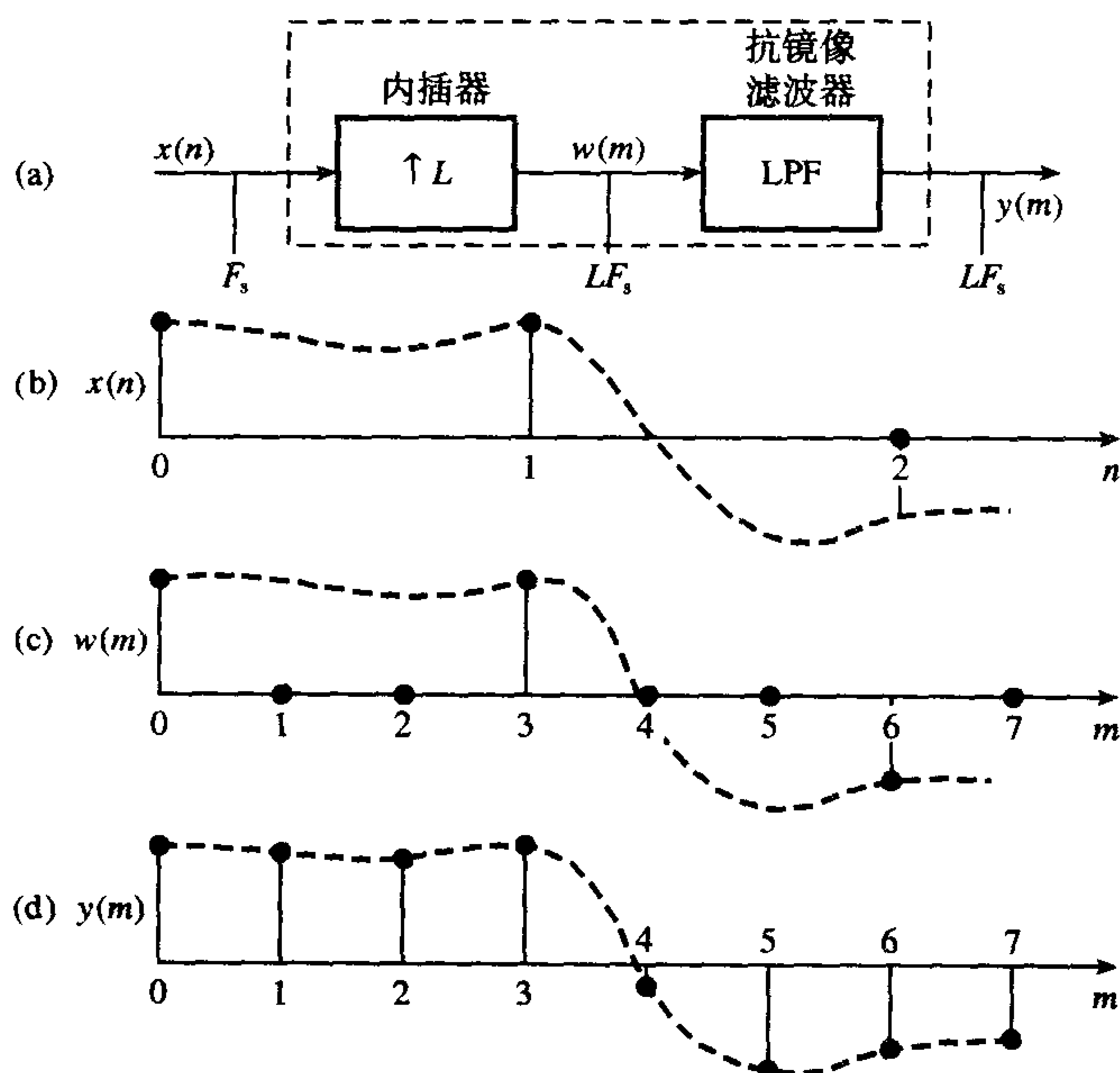


图 9.3 一个信号按整数因子$L = 3$的内插过程的时域描述。注意到对于$x(n)$的每个抽样，$y(m)$中获得 3 个输出抽样

图 9.4 给出了内插过程的频域描述。$X(f)$、$W(f)$和$Y(f)$分别是信号$x(n)$、$w(m)$和$y(m)$的频率响应。$H(f)$是去镜像滤波器的幅度响应，这个滤波器用来去除$W(f)$的虚线部分所代表的镜像分量。现在应该指出，尽管某些聪明的读者可能早已怀疑，抽取和内插过程是互相对偶的，即一个是另一个的逆过程。这种对偶的性质表明内插滤波器完全可以从等效的抽取滤波器推得；反之亦然。

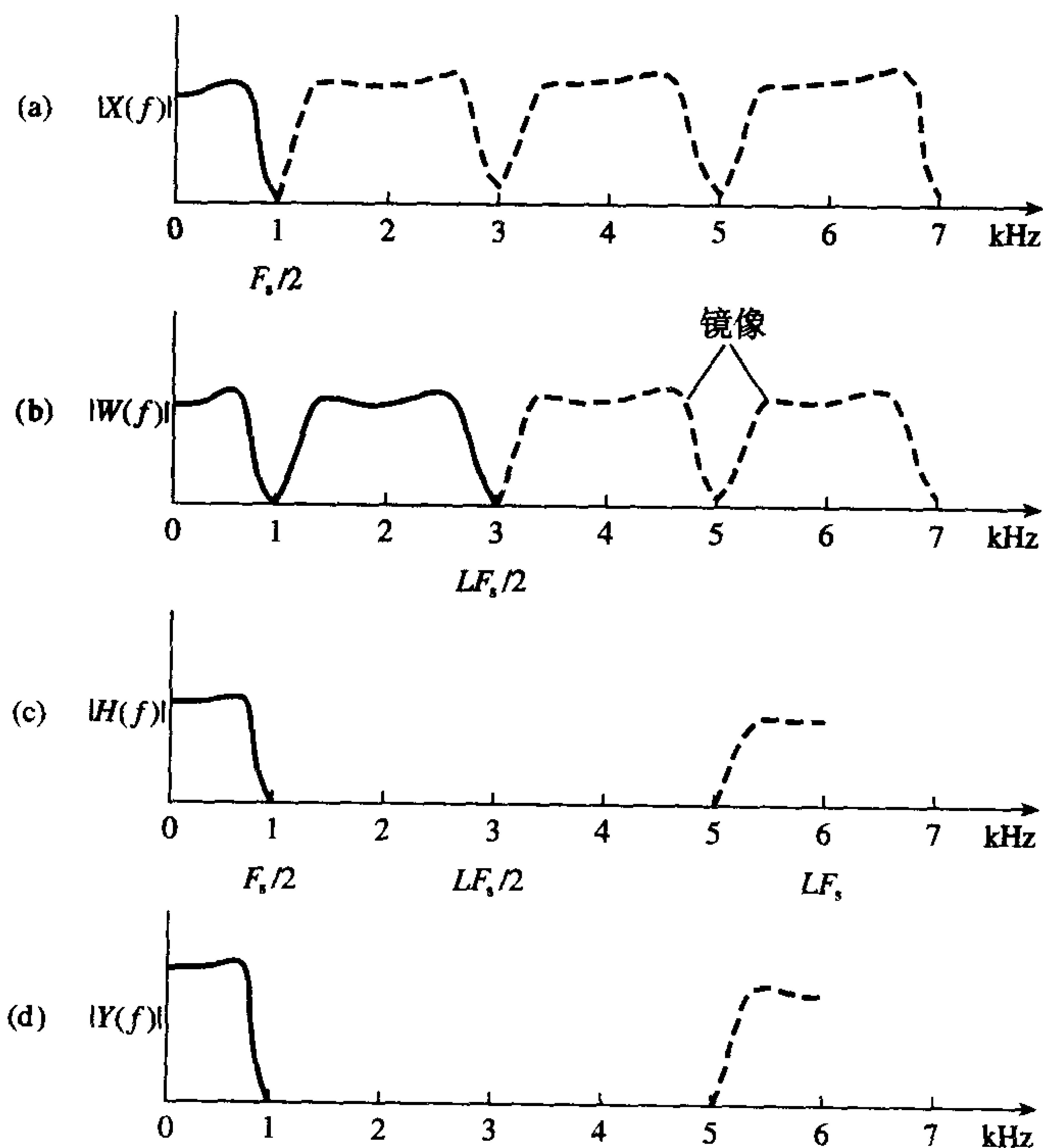


图 9.4 一个信号从 2 kHz 到 6 kHz 内插过程的频域描述

9.2.3　抽样率按非整数因子变换

在某些场合，常常需要对抽样率按一个非整数因子改变。一个例子是在数字音频应用中，需要将数据从一个存储系统传输到另一个，而这两个系统使用不同的抽样率，可能是为了防止某些非法拷贝。比如将抽样率为44.1 kHz的光盘（CD）数据传输到抽样率为48 kHz的数字录音带上（DAT），这需要将CD数据的抽样率按非整数因子48/44.1增加。

在实际操作中，这种非整数因子可以用一个有理数或者两个整数L和M之比来表示，这里L和M是使L/M尽量接近所希望因子的整数。抽样频率的变换通过先将数据按因子L内插，再按因子M抽取来达到（参见图9.5(a)）。将内插过程放在抽取之前是很有必要的，这可以防止抽取过程把那些需要的频率分量丢弃。对于前面CD-DAT的例子，抽样率48/44.1可以先将数据按$L=160$内插，再按$M=147$抽取来得到，即先将CD数据抽样率按$L=160$提高到7056 kHz，再按$M=147$降低到48 kHz。

图9.5(a)中的两个线性相位滤波器（LPF）$h_1(k)$和$h_2(k)$，由于二者级联且具有共同的抽样频率，可以合并成一个单独的滤波器，构成了图9.5(b)的通用抽样率转换器。如果$M>L$，则操作是一个按非整数因子抽取的过程，而$M<L$则是一个内插过程。如果$M=1$，则通用抽样率转换器变成了先前的按整数因子内插的过程，而$L=1$则变成整数抽取过程。

图9.5(c)描述了按3/2内插的时域过程。通过在$x(n)$的每个抽样点间插入2个零值抽样，抽样率先增加了3倍，再低通滤波产生$v(i)$。滤波输出数据的每2个抽样点只保留1个，得到抽样率降低一半的最后输出$y(m)$。图9.6给出了按3/2内插过程的频域描述。输入信号$x(n)$具有2 kHz的抽样率，先被增加3倍到6 kHz，滤波去除可能造成混叠的镜像频率分量，再降低抽样率一半到3 kHz。

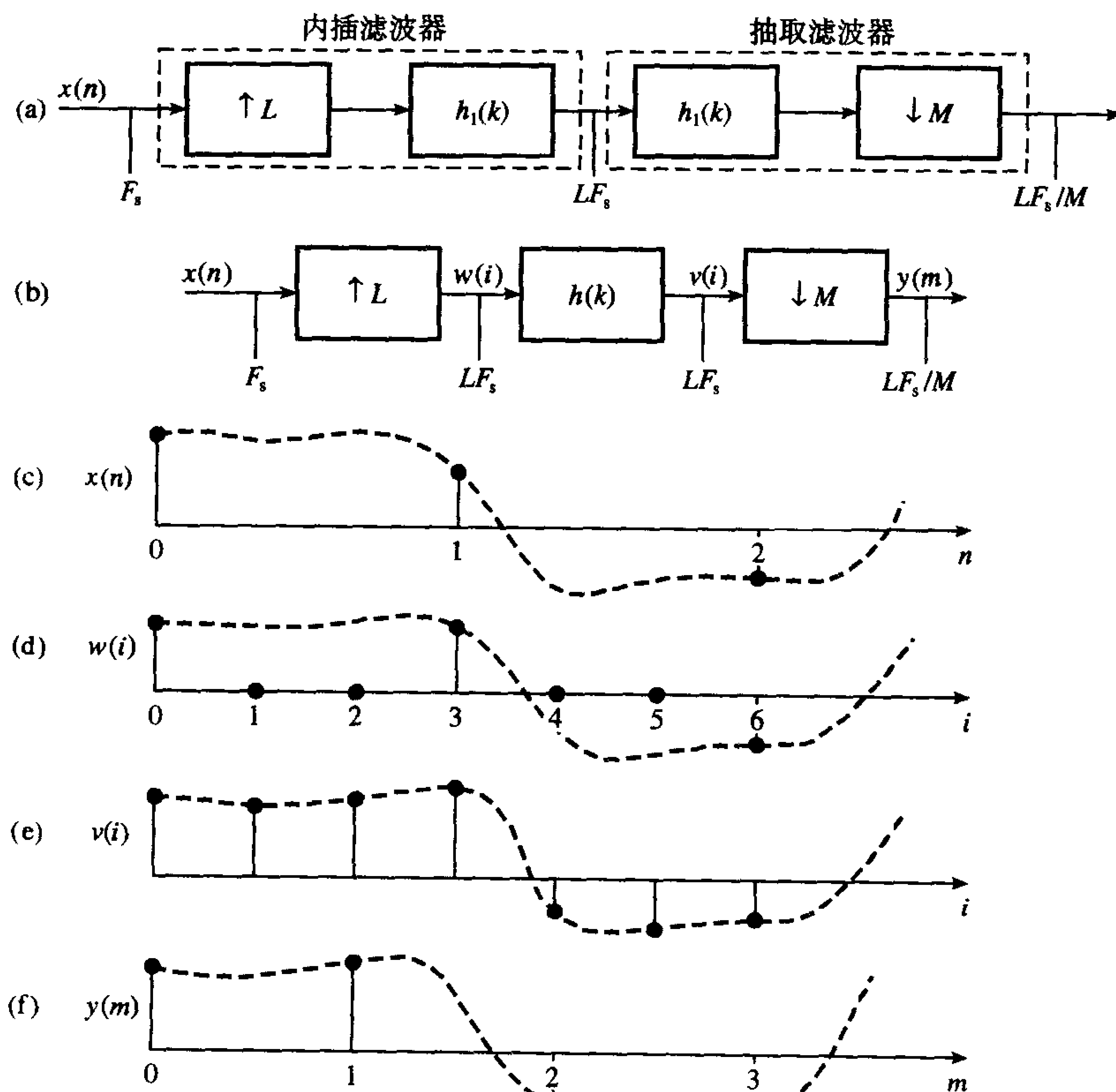


图9.5　一个信号按有理数因子（$L=3$，$M=2$）内插的时域描述

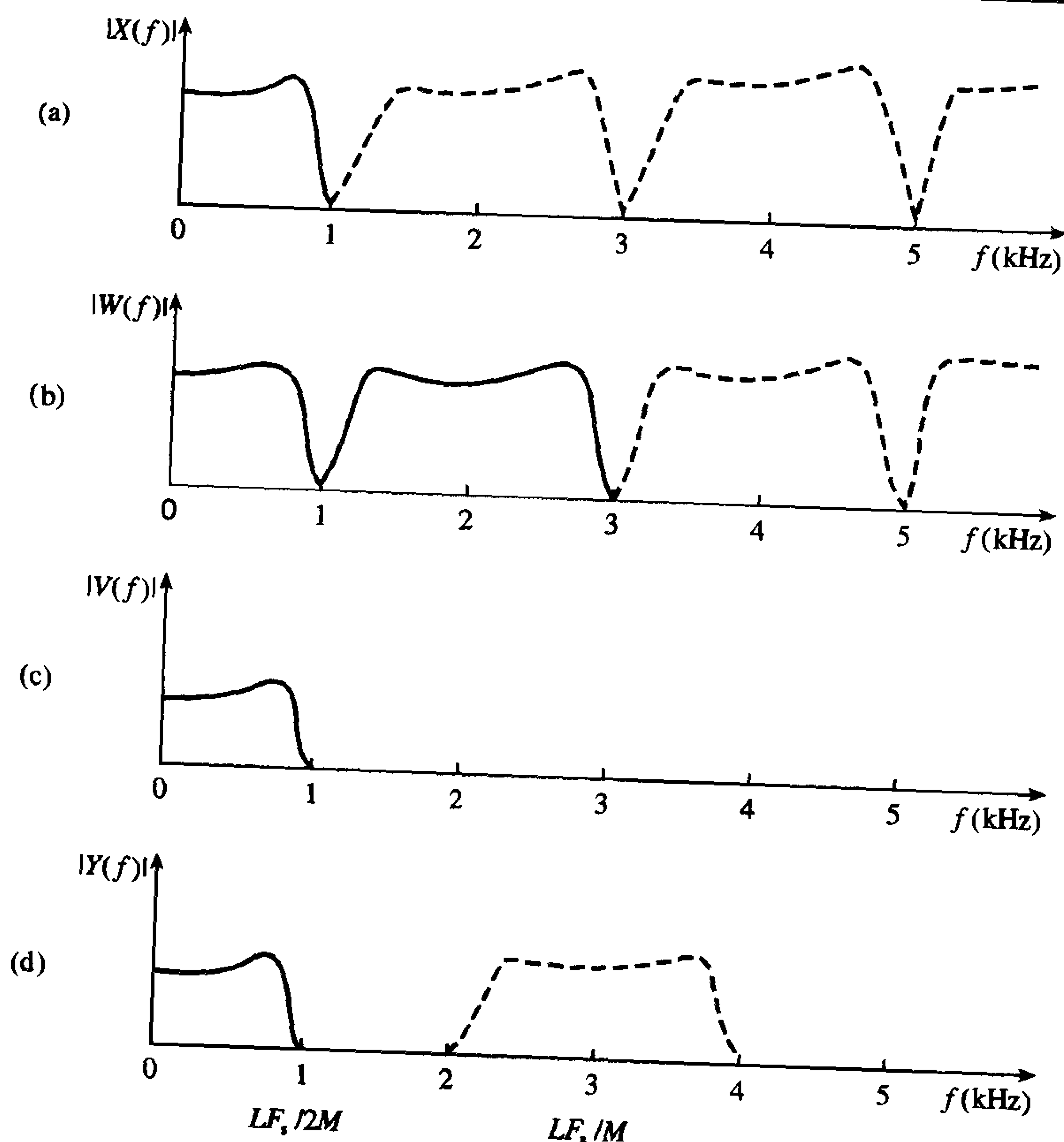


图 9.6　一个信号从 2 kHz 按 3/2 因子内插的频域描述。信号抽样率先增加 3 倍内插到 6 kHz (a)；再带限滤波以防止混叠(c)，最后抽取将抽样率降到 3 kHz

例 9.1　一个抽样频率为 2.048 kHz 的信号 $x(n)$，按因子 32 抽取产生抽样频率为 64 Hz 的信号。感兴趣的信号频带为 0 ~ 30 Hz。抗混叠数字滤波器应满足以下指标：

通带偏差	0.01 dB
阻带衰减	80 dB
通带	0 ~ 30 Hz
阻带	32 ~ 64 Hz

应防止信号的 30 ~ 32 Hz 的频带分量所产生的混叠。请设计一个合适的一阶抽取滤波器。

解：

图 9.7 给出了一阶抽取滤波器的框图和低通抗混叠滤波器的频响要求。根据题目及频响指标，我们可以设计以下滤波器参数：

$$\Delta f = (32-30)/2048 = 9.766\times 10^{-4}$$

$\delta_p = 0.001\,15$,　　由 $20\log(1+\delta_p) = 0.01$ dB 得出

$\delta_s = 0.0001$,　　由 $-20\log(\delta_s) = 80$ dB得出

单级抽取滤波器的滤波器系数数目的估计为（参见第 7 章）

$$N \approx \frac{D_\infty(\delta_p, \delta_s)}{\Delta f} - f(\delta_p, \delta_s)\,\Delta f + 1 \tag{9.3}$$

其中 Δf 是按抽样频率归一化后的过渡带宽，

$$D_\infty(\delta_p, \delta_s) = (\log_{10}\delta_s)[a_1(\log_{10}\delta_p)^2 + a_2(\log_{10}\delta_p) + a_3] + a_4(\log_{10}\delta_p)^2 + a_5(\log_{10}\delta_p) + a_6$$

$$f(\delta_p, \delta_s) = 11.012\,17 + 0.512\,44(\log_{10}\delta_p - \log_{10}\delta_s)$$

$$a_1 = 5.309\times10^{-3};\quad a_2 = 7.114\times10^{-2};$$

$$a_3 = -4.761\times10^{-1};\quad a_4 = -2.66\times10^{-3};$$

$$a_5 = -5.941\times10^{-1};\quad a_6 = -4.278\times10^{-1}.$$

δ_p 和 δ_s 分别是通带和阻带波纹或偏差。

利用 9.3 式中 δ_p、δ_s 和 Δf 的值，我们得到 $N = 3947$，显然 N 太大。实际上，没有任何可行的滤波器设计方法来得到这样一个滤波器，因为近似误差将难以接受。对于所有的实际情况，为一阶抽取滤波器来设计低通滤波器都是不可行的。这个例子说明需要采用某种变通的方法，更有效地进行抽样率的变换，尤其是抽样率的变化较大时。下一节将进行有关的讨论。

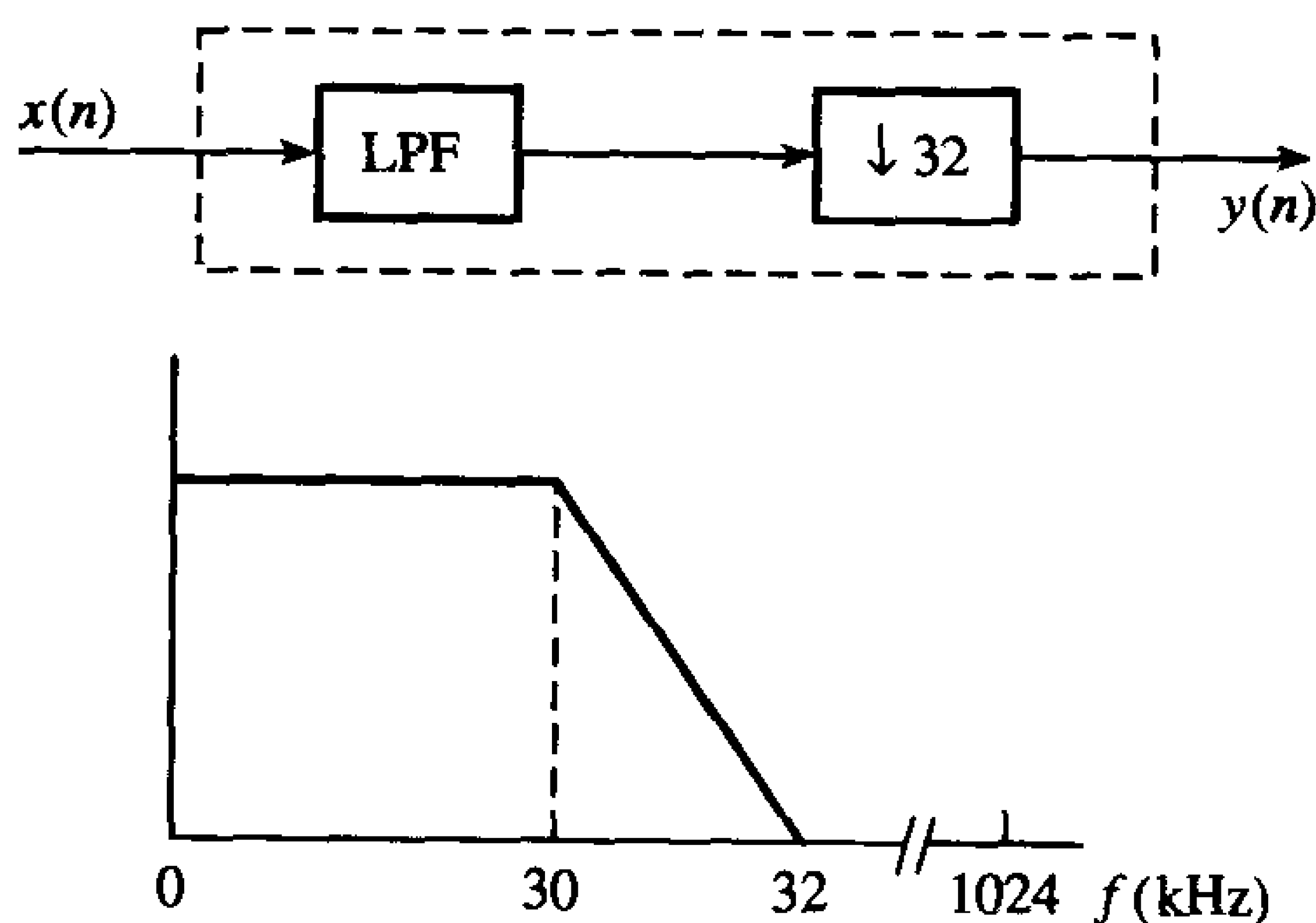


图 9.7　满足例 9.1 的一阶抽取滤波器

9.2.4　抽样率变换的多级方法

在上一节里，抽样率的变换是通过抽取或内插因子一步到位的。当抽样率的变化较大时，采用二级或多级可以更有效地进行抽样率转换。实际上，大多数多抽样率系统使用了多级方法，它能渐进地降低或增加抽样率，带来显著放宽抗混叠滤波器或去镜像滤波器的技术性能指标的好处。

图 9.8 给出了一个 I 级抽取过程的框图。总的抽取因子 M 可以表示成各级抽取因子的乘积：

$$M = M_1 M_2 \ldots M_I \tag{9.4}$$

其中 M_I 代表各级抽取因子，是一个整数。各线框内是互相独立的抽取滤波器。如果 $M \gg 1$，多级结构的抽取滤波器能大大降低总的计算量和存储要求，减轻滤波器的设计难度，并且滤波器的总有限字长效应也比较小。

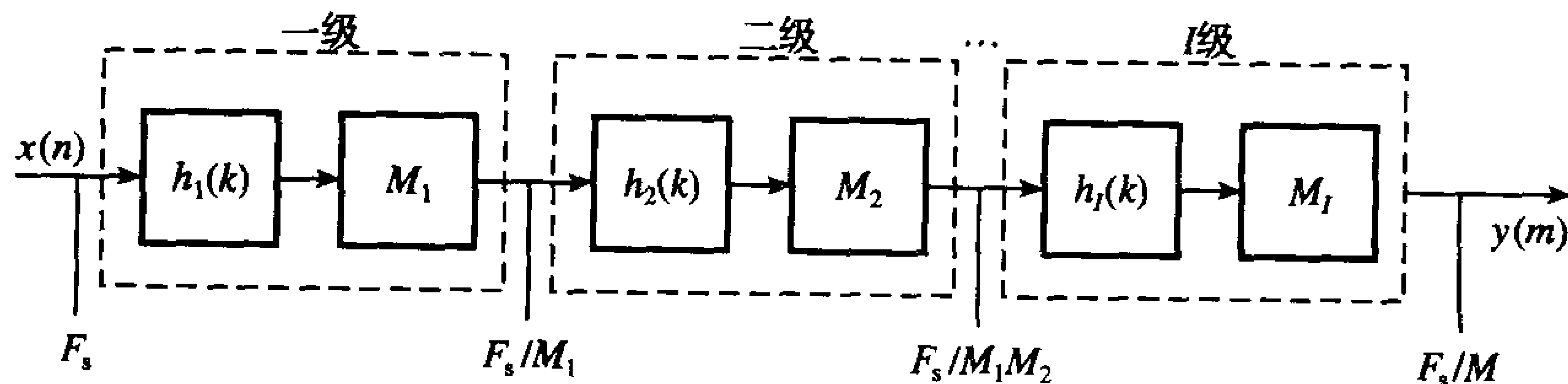


图 9.8　多级抽取过程

这些优点的代价是设计和实现系统的难度增加。在我们介绍完设计方法后，将给出多个例子来说明多级法。

9.3 设计实际的抽样率变换器

设计一个实用的多级抽样率变换器可划分为下面4个步骤：

(1) 确定总的抗混叠滤波或去镜像滤波指标；
(2) 确定最佳的抽取或内插级数，以便于更有效地实现；
(3) 确定每级的抽取或内插因子；
(4) 设计每级需要的滤波器。

9.3.1 滤波器指标

根据前面的论述，在抽样率变换过程中对抗混叠或去镜像数字滤波器的需求是明确的。实际上，多抽样率系统的总体性能严重依赖于所使用滤波器的类型和品质。FIR和IIR滤波器都可以用于抽取或内插过程，但通常我们使用FIR滤波器。

与普通DSP不同，在多抽样率处理中，FIR滤波器的计算效率与IIR滤波器大致相当，有时还要好一些（Crochiere and Rabiner, 1975, 1976, 1983）。另外，FIR滤波器具有许多优点（参见第6章和第7章），如线性相位响应、对有限字长效应不很敏感及易实现等。出于这些考虑，本章将只讨论FIR滤波器。第7章所介绍的计算FIR滤波器参数的方法都可以用于本章的多抽样率系统设计。在实践中，最普遍应用的是最优和半带滤波器。

对于抽取过程，为避免抽样率降低引起频谱混叠，总的滤波器指标为

通带	$0 \leqslant f \leqslant f_p$	(9.5a)
阻带	$F_s/2M \leqslant f \leqslant F_s/2$	(9.5b)
通带波纹	δ_p	(9.5c)
阻带波纹	δ_s	(9.5d)

其中$f_p < F_s/2M$，F_s是原始抽样频率。一般来说，f_p是原始信号中我们感兴趣的最高频率。

对于内插，去镜像滤波器必须带限内插零值后的数据到$F_s/2$或更小，将所有无用信息滤除。尽管根据抽样定理，增加抽样率到LF_s后的最高有效频率是$LF_s/2$，但仍应该带限至$F_s/2$这个原信号$x(n)$的最高有效频率。对于内插过程，总的滤波器指标为

通带	$0 \leqslant f \leqslant f_p$	(9.6a)
阻带	$F_s/2 \leqslant f \leqslant LF_s/2$	(9.6b)
通带波纹	δ_p	(9.6c)
阻带波纹	δ_s	(9.6d)

其中$f_p < F_s/2$。在通带中应加入一个增益L以补偿内插过程对信号的幅度衰减。

9.3.2 单级的滤波器指标

尽管也可以使用加窗法，等波纹（最优）滤波器通常用于抽样率变换中。图9.9(a)给出了一个等波纹低通滤波器的取值范围。

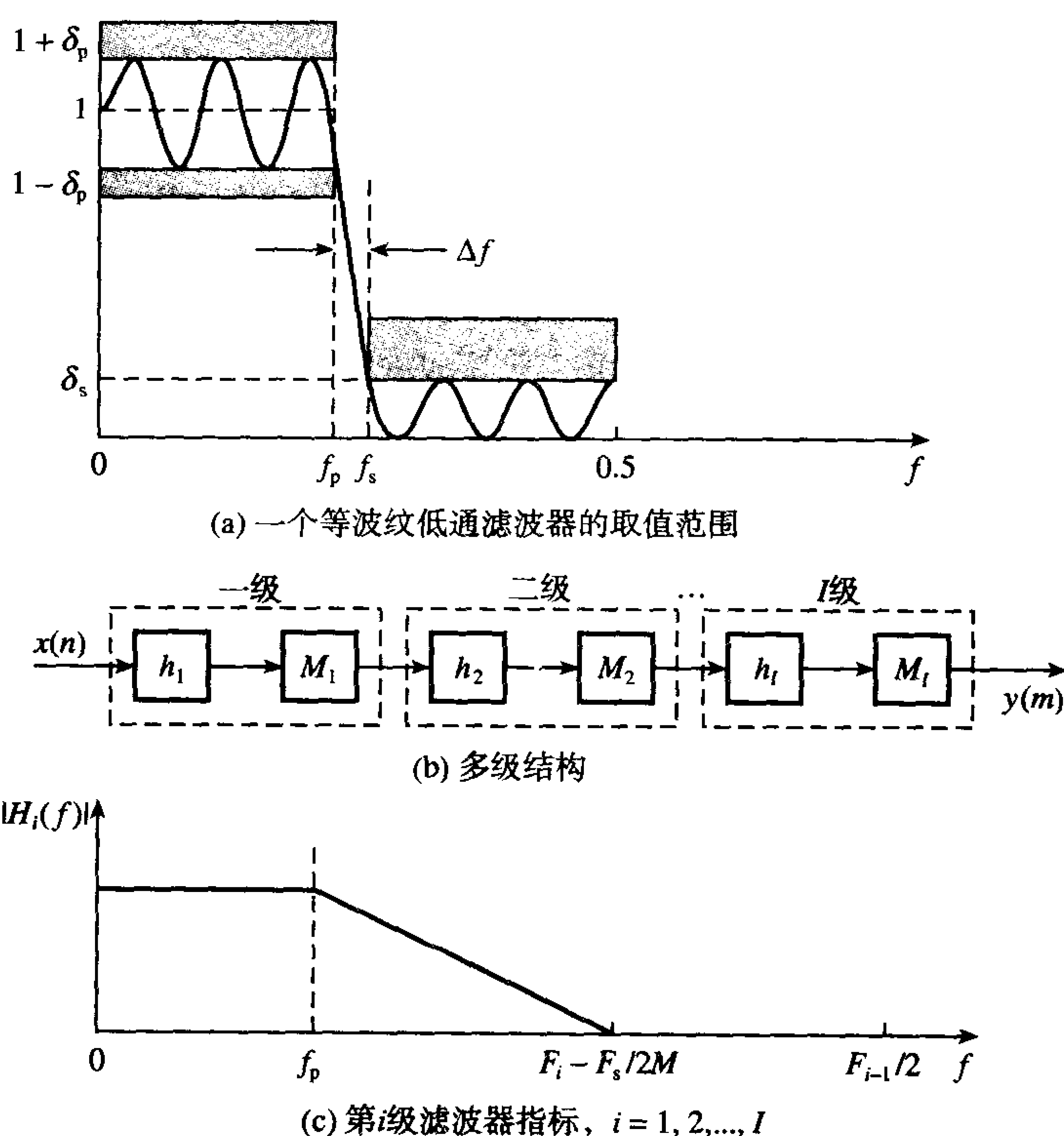


图9.9　一个等波纹低通滤波器的取值范围，以及一个多级抽取滤波器和其各级指标

对于一个多级抽取滤波器（参见图9.9(b)），为保证总的滤波效果，各级滤波器指标分别为（参见图9.9(c)）

$$\text{通带}\quad 0 \leqslant f \leqslant f_{\mathrm{p}} \tag{9.7a}$$

$$\text{阻带}\quad (F_i - F_{\mathrm{s}}/2M) < f < F_{i-1}/2, \quad i = 1, 2, \ldots, I \tag{9.7b}$$

$$\text{通带波纹}\quad \delta_{\mathrm{p}}/I \tag{9.7c}$$

$$\text{阻带波纹}\quad \delta_{\mathrm{s}} \tag{9.7d}$$

$$\text{滤波器长度}\quad N \approx \frac{D_{\infty}(\delta_{\mathrm{p}}, \delta_{\mathrm{s}})}{\Delta f_i} - f(\delta_{\mathrm{p}}, \delta_{\mathrm{s}})\,\Delta f_i + 1 \tag{9.7e}$$

其中F_i、N_i和Δf_i分别是第i级抽取滤波器的输出抽样频率、滤波器长度和归一化过渡带宽。参数$D_{\infty}(\delta_{\mathrm{p}}, \delta_{\mathrm{s}})$和$f(\delta_{\mathrm{p}}, \delta_{\mathrm{s}})$的含义与9.3式相同。第$i$级输出抽样频率由下式给出：

$$F_i = F_{i-1}/M_i, \quad i = 1, 2, \ldots, I \tag{9.8}$$

其中M_i是单级抽取因子。初始及最终的抽样率分别为F_0和F_I。与前面的论述相对照，$F_0 = F_{\mathrm{s}}$，$F_I = F_{\mathrm{s}}/M$。

对于多级抽取，各级的低通偏差应保证总的通带偏差δ_{p}。各级的阻带偏差应与总的阻带偏差一致，这是由于当信号连续经过各级时，阻带分量处于不断的衰减之中。对于一个单级抽取滤波器，滤波器指标与9.5式一致。

9.3.3 确定多级结构的级数和抽取因子

采用多级结构设计比单级结构大幅度减少了对计算和存储的需求。减少的程度取决于多级结构的级数和单级的抽取因子。主要的问题在于确定最优的级数I和各级抽取因子M_i。最优级数应能产生最小的计算量，例如每秒的乘法次数（MPS）或对滤波系数的总存储需求（TSR）：

$$\text{MPS} = \sum_{i=1}^{I} N_i F_i \tag{9.9a}$$

$$\text{TSR} = \sum_{i=1}^{I} N_i \tag{9.9b}$$

其中N_i是第i级滤波器系数的个数，且我们忽略了任何滤波器系数的对称性。

级数I和抽取因子的选择并不是一个简单的问题。然而，在实际应用中，多级结构的级数很少超过3或4。另外，对于一个给定的抽取因子M值，其整因数集是很有限的。这导致了一种可行的确定所有可能的M因数的方法，包括所有的M_i值和对应的MPS或总存储需求，并在其中选择最有效或满足要求的解。该方法的计算步骤总结在表9.1中，一个C语言实现程序在指导手册的CD中，结果数据表由DeFatta等人于1988年发表。

表9.1 寻找最优I和M_i值的方法

- 指定所有滤波参数（$F_s, M, f_p, f_s, \delta_s, \delta_p$）
- 对于I（$I = 1, 2, \ldots, I_{max}$）的每个取值，求所有可能的抽取因子等于$M$的整因数集合
- 根据每个整因数集的抽取因子确定滤波器指标，并利用9.9式计算MPS和TSR
- 对于每个I值，根据存储需求范围内计算最有效的原则来选择抽取因子
- 选择最有效或满足要求的解

总体上看，最优的MPS或TSR满足以下关系（Crochiere and Rabiner, 1975, 1976）

$$M_1 > M_2 > \ldots M_I \tag{9.10}$$

其中M_i是连续的。然而，当M为整数时，9.10式并不总能满足。例如，如果$I=3$且$M=32$（参见对例9.3的讨论）。

对$I=2$，即二级抽取，最小化TSR得到的抽取因子最优值为

$$\text{M}_{1_{opt}} = \frac{2M}{2 - \Delta f + (2M\Delta f)^{1/2}} \tag{9.11a}$$

$$\text{M}_{2_{opt}} = \frac{M}{M_{1_{opt}}} \tag{9.11b}$$

对$I>2$，没有简单的闭式表达存在，必须利用计算机辅助最优化程序或者表9.1来寻找最优的各级抽取因子M_i。

9.3.4 图解设计例题

例9.2

(a) 利用三级抽取滤波器来将抽样频率从3072 kHz降低到48 kHz的框图由图9.10给出。假定各级抽取因子分别为8、4和2，指出各级输出的抽样频率。

(b) 假定(a)中的抽取滤波器满足以下条件：

输入抽样频率F_s	3072 kHz
抽取因子M	64

通带偏差	0.01 dB
阻带衰减	60 dB
感兴趣频带	0 ~ 20 kHz

确定各级抽取的带边沿频率。

(c) 假定一个抽取滤波器的输入和输出抽样率分别为 3072 kHz 和 48 kHz：

(i) 写出总的抽取因子；

(ii) 写出抽取因子的所有可能的整因数集（按降序），并假定只有二级抽取；

(iii) 重复(ii)但假定三级抽取；

(iv) 重复(ii)但假定四级抽取。

(d) 对(a)中的抽取滤波器，计算总的每秒乘法次数 MPS 及各种滤波器长度下的总存储需求。

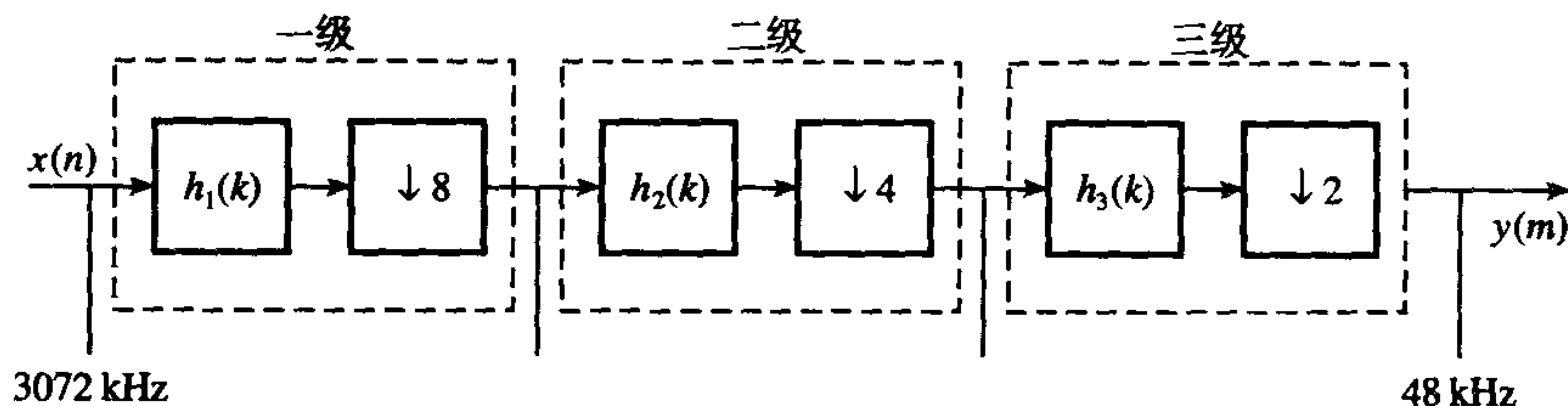


图 9.10　例 9.2 的框图

解：

(a) 在第一级，抽样率从 3072 kHz 按抽取因子 8 被降低到 384 kHz。第二级抽样率从 384 kHz 按抽取因子 4 被降低到 96 kHz。第三级抽样率从 96 kHz 按抽取因子 2 被降低到 48 kHz。

(b) 3 个抽取（抗混叠）滤波器的通带边沿频率是相同的（即 24 kHz），用以保护感兴趣的频带。阻带边沿频率是不同的，可以利用抽样率的差别。带阻频率根据以下关系式计算（参见图 9.9(c) 和 9.7b 式）：

$$f_{si} = F_i - \frac{F_s}{2M},\quad 1, 2, 3$$

其中

F_i = 各级输出抽样频率

F_s = 系统基本抽样率

f_{si} = 各级阻带边沿频率

对第一级，f_{s1} = 384−3072/(2 × 64) = 360 kHz。因此，第一级抽取滤波器的带边沿频率分别为：0、20 kHz、360 kHz 和 1536 kHz（该级的奈奎斯特频率，即 3072 kHz/2）。

对第二级，f_{s2} = 96−3072/(2 × 64) =72 kHz。带边沿频率分别为：0、20 kHz、72 kHz 和 192 kHz。

对第三级，f_{s3} = 48−3072/(2 × 64) = 24 kHz。抗混叠滤波器的带边沿频率分别为：0、20 kHz、24 kHz 和 48 kHz。

(c) (i) 总的抽取因子为 3072/48 = 64。

(ii) 二级抽取可能的整因数集为（按降序）

32 × 2

16 × 4

8 × 8

(ii) 三级抽取可能的整因数集为（按降序）

$16 \times 2 \times 2$

$8 \times 4 \times 2$

(iv) 四级抽取可能的整因数集为

$4 \times 4 \times 2 \times 2$

(d) 假定抽取因子为 $8 \times 4 \times 2$ 且对应的滤波器长度分别为 N_1、N_2 和 N_3，则总的 MPS 为

$$N_1 \times F_1 + N_2 \times F_2 + N_3 \times F_3 = N_1 \times 384 \times 10^3 + N_2 \times 96 \times 10^3 + N_3 \times 48 \times 10^3$$

总的 TSR 为

$$N_1 + N_2 + N_3$$

例9.3 通过抽取使信号 $x(n)$ 的抽取率下降，从 96 kHz 到 1 kHz。抽取后感兴趣的最高频率为 450 Hz。假定使用一个最优 FIR 滤波器，其总的通带波纹 $\delta_p = 0.01$，总的阻带偏差 $\delta_s = 0.001$。设计一个有效的抽取滤波器。

解：

我们首先寻找对每个 $I = 1, 2, 3, 4$ 的最有效设计，再比较这些设计来选择最好的。

(1) 首先让我们考虑一级设计（$I = 1$）。框图和滤波器指标在图 9.11(a) 中给出。

(2) 接下来我们考虑二级设计。利用前文中的设计程序，$I = 2$ 时抽取因子的最优整因数是 $M_1 = 32$，$M_2 = 3$。二级系统，包括其滤波器指标，在图 9.11(b) 中给出。在第一级，抽样率按抽取因子 32 降低到 3 kHz，在第二级，进一步按抽取因子 3 降低到 1 kHz。

(3) 对三级的情况（$I = 3$），再考虑总存储需求时，抽取因子的最优整因数是 $M_1 = 8$，$M_2 = 6$，$M_3 = 2$。该系统在图 9.11(c) 中给出。

(4) 对四级设计，最优整因数是 $M_1 = 4$，$M_2 = 4$，$M_3 = 3$，$M_4 = 2$。系统及滤波器指标参见图 9.11(d)。

最终结果总结如下：

I	N_1	N_2	N_3	N_4	M_1	M_2	M_3	M_4	MPS	TSR
1	4881	–	–	–	96	–	–	–	48 881 000	4881
2	131	167	–	–	32	3	–	–	560 000	298
3	25	34	117	–	8	6	2	–	485 000	176
4	11	13	17	120	4	4	3	2	496 000	161

显然，总体来说，多级设计比单级设计能在很大程度上降低了计算量和存储需求。这种降低来源于前级滤波器带来的巨大变化（尽管抽样率仍比较高），导致了较小的 N 值（滤波器系数个数）。

在比较多级设计的有效性时，我们发现计算量（MPS）和存储量（TSR）的降低在从一级到二级时最大。从二级到三级或者三级到四级时，存储量减少，但不明显，计算量在第二级和第三级之间也有一定的降低，但并不明显。从三级到四级，计算量（MPS）实际上增加了。总的来看，$I = 3$ 是最有效的实现结构，还应考虑随着 I 的增加其实现难度也在增加。因为在实际应用中，最终选择还需在考虑硬件成本和软件复杂度后做出。

评论 根据 Crochiere 和 Rabiner（1975, 1976）的论述，当 M_i（$i = 1, 2, \ldots, I$）是连续变量时，最优 M_i 值满足条件 $M_1 > M_2 > \ldots M_I$。

另外，最小化存储需求的 M_i 值同样能最小化计算量。然而，M_i 值受限于必须是整数，通常达不到最小化条件。由于这个原因，我们的设计程序实际上只计算所有可能的整数因子解。再通过观察来选择最好的方案。

在考虑MPS或TSR下，最有效的解可能在某级包含一个过大（从而难以实现）的N值。另一套抽取因子集或增加总的级数可能带来希望的N值降低，代价是可能在其他级上增加滤波器长度。通过罗列所有的解，从中比较来得到折中的选择。

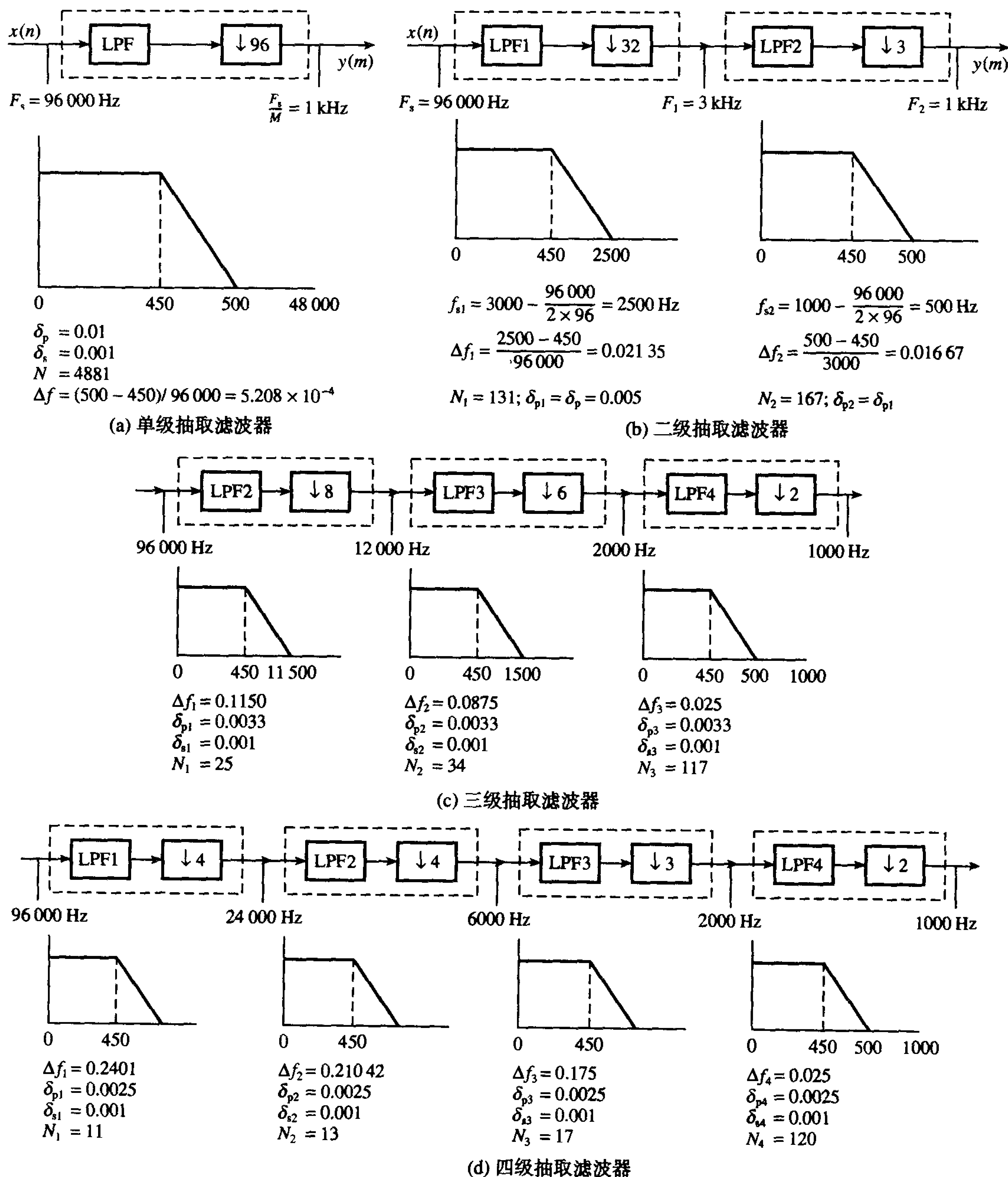


图 9.11　对应例 9.3 的图示

例 9.4　在框图层次上设计一个二级抽取滤波器，将一个音频信号按因子 30 降低其抽样率，满足表 9.2 的需求。

你的答案必须包括所有适合的抽取因子，以及详细分析它们的计算和存储需求来检验你的选择。确定各级抽取的输入/输出抽样频率，下面给出满足设计要求的抽取滤波器指标：

带沿频率
归一化过渡带宽
通带和阻带波纹
滤波器长度

你可以假定滤波器都是直接FIR结构，长度由表9.2给出。

表9.2 二级抽取滤波器指标

输入抽样频率 F_s	240 kHz
感兴趣的最高频率	3.4 kHz
通带波纹系数 δ_p	0.05
阻带波纹系数 δ_s	0.01
滤波器长度，$N = \dfrac{-10\log(\delta_p\delta_s) - 13}{14.6\Delta f} + 1$	
其中 Δf = 归一化过渡带宽	

解：

抽取滤波器应具有形如图9.12的结构框图。

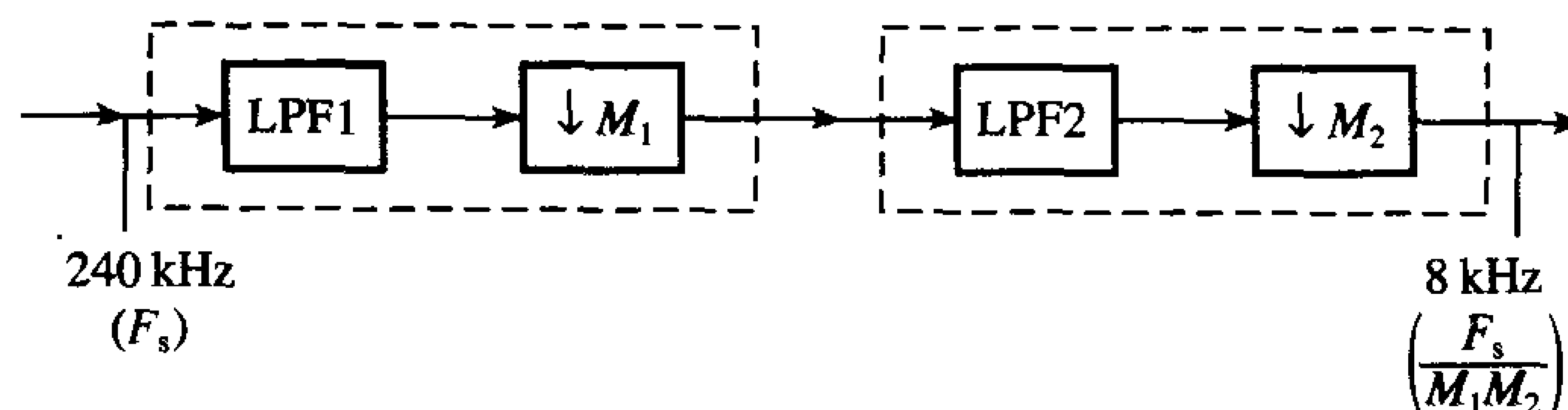


图9.12 一个二级抽取滤波器

假设是整数抽取因子，则可能的整因数对是（考虑到计算效率，第一级永远取最大的因数）

15 × 2

10 × 3

6 × 5

为了决定使用哪一对抽取因子，必须进行计算复杂度分析。

(i) 对 $M_1 = 15$ 和 $M_2 = 2$，抽样率先按因子15下降到16 kHz，再按因子2下降到8 kHz。二级抽取滤波器的参数为

第一级：带边沿频率：3.4 kHz和12 kHz $\left(16 - \dfrac{240}{2\times 30}\right)$

$$\Delta f = \frac{12 - 3.4}{240} = 0.0358$$

$$\delta_p = \frac{0.05}{2} = 0.025;\ \delta_s = 0.01;\ N_1 = 45$$

第二级：带边沿频率：3.4 kHz和4 kHz

$$\Delta f = 0.0375$$

$$\delta_p = \frac{0.05}{2} = 0.025;\ \delta_s = 0.01;\ N_2 = 43$$

两个复杂度指标分别是每秒乘法次数（MPS）和总存储需求（TSR）。为了执行的效率，选择抽取滤波器结构使各级滤波操作在低抽样率下进行，结果是

$$\text{MPS} = (45 \times 16 + 43 \times 8) \times 10^3 = 1064 \times 10^3; \text{存储量} = 88$$

(ii) 对 $M_1 = 10$ 和 $M_2 = 3$，抽样率先按因子 10 下降到 24 kHz，再按因子 3 下降到 8 kHz。二级抽取滤波器的参数为

第一级：带边沿频率：3.4 kHz 和 20 kHz $\left(24 - \dfrac{240}{2 \times 30} = 20\,\text{kHz}\right)$

$$\Delta f = \frac{20 - 3.4}{240} = 0.0691; \delta_p = \frac{0.05}{2} = 0.025; \delta_s = 0.01; N_1 = 23.81 \approx 24$$

第二级：带边沿频率：3.4 kHz 和 4 kHz（8–4 = 4 kHz）

$$\Delta f = \frac{4 - 3.4}{24} = 0.025; \delta_p = \frac{0.05}{2} = 0.025; \delta_s = 0.01; N_2 = 64$$

这时的两个复杂度指标分别为

$$\text{MPS} = (24 \times 24 + 64 \times 8) \times 10^3 = 1088 \times 10^3;\ \text{存储量} = 88$$

(iii) 最后一个可能的抽取因子对是 $M_1 = 6$ 和 $M_2 = 5$。同样的分析有

第一级：带边沿频率：3.4 kHz 和 36 kHz

$$\Delta f = 0.1358;\ \delta_p = \frac{0.05}{2} = 0.025; \delta_s = 0.01; N_1 = 13$$

第二级：带边沿频率：3.4 kHz 和 4 kHz

$$\Delta f = 0.015;\ \delta_p = \frac{0.05}{2} = 0.025; \delta_s = 0.01; N_2 = 106$$

$$\text{MPS} = 1368 \times 10^3;\ \text{存储量} = 119$$

三种情况下的复杂度指标总结在表9.3中。比较计算和存储的复杂度，我们可以看出最好的抽取因子对是 $M_1 = 15$ 和 $M_2 = 2$。

表 9.3　计算（MPS）和存储（TSR）复杂度

抽取因子	MPS	TSR
$M_1 = 15; M_2 = 2$	1064×10^3	88
$M_1 = 10; M_2 = 3$	1088×10^3	88
$M_1 = 6; M_2 = 5$	1368×10^3	119

9.4　抽样率变换器——抽取滤波器的软件实现

图9.13(a)给出了抽取滤波器的一个简单框图解。其中 $h(k)$ 是一个抗混叠滤波器。假定采用一种直接实现结构（即使用延迟线滤波），滤波器的输出 $w(n)$ 和输入 $x(n)$ 之间的关系为

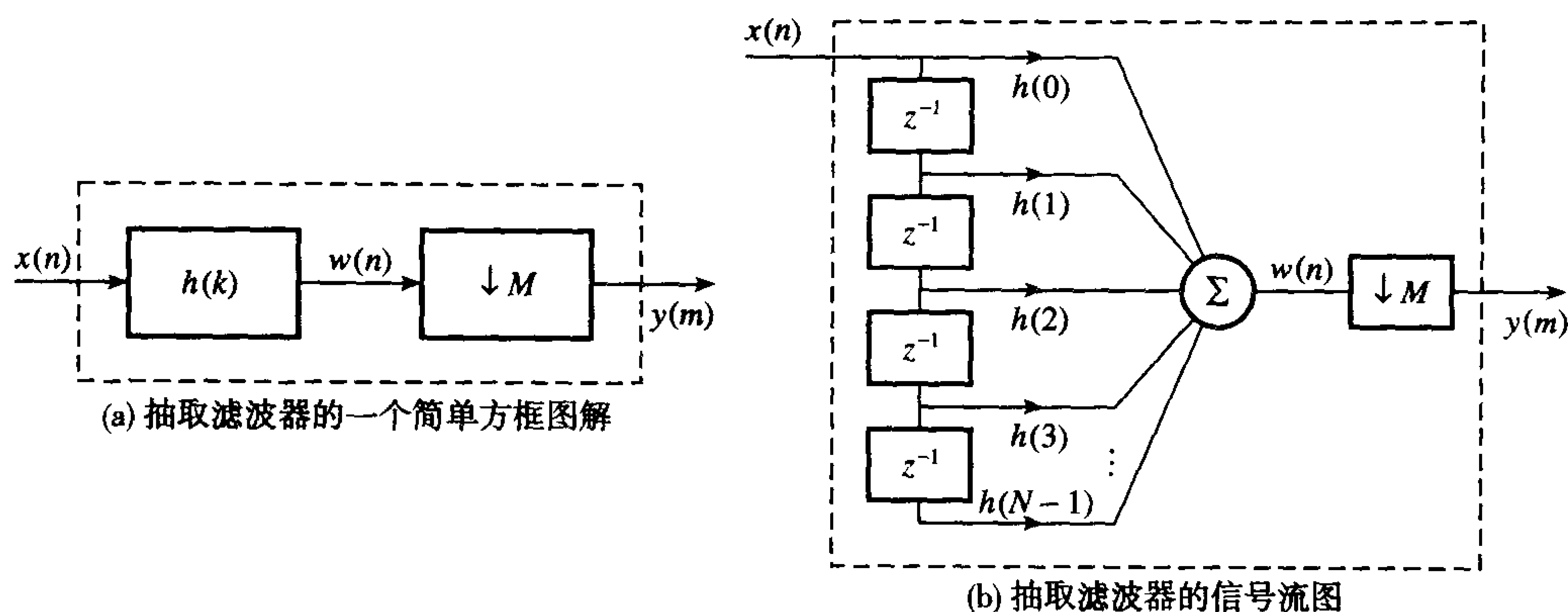
$$w(n) = \sum_{k=0}^{N-1} h(k)x(n-k) \tag{9.12a}$$

其中 N 是 FIR 滤波器系数的个数。抽取滤波器输出 $y(m) = w(mM)$，代入 9.12a 式得到抽取方程：

$$y(m) = \sum_{k=0}^{N-1} h(k)x(Mm-k) \tag{9.12b}$$

抽取滤波器的信号流图参见图9.13(b)。输入 $x(n)$ 加到延迟时间为一个抽样时间的延迟线。$x(n)$ 每 M 个抽样加到延迟线，计算一个输出 $y(m)$。也就是保留 $w(n)$ 的第一个抽样，抛弃接下来的 M–1 个抽

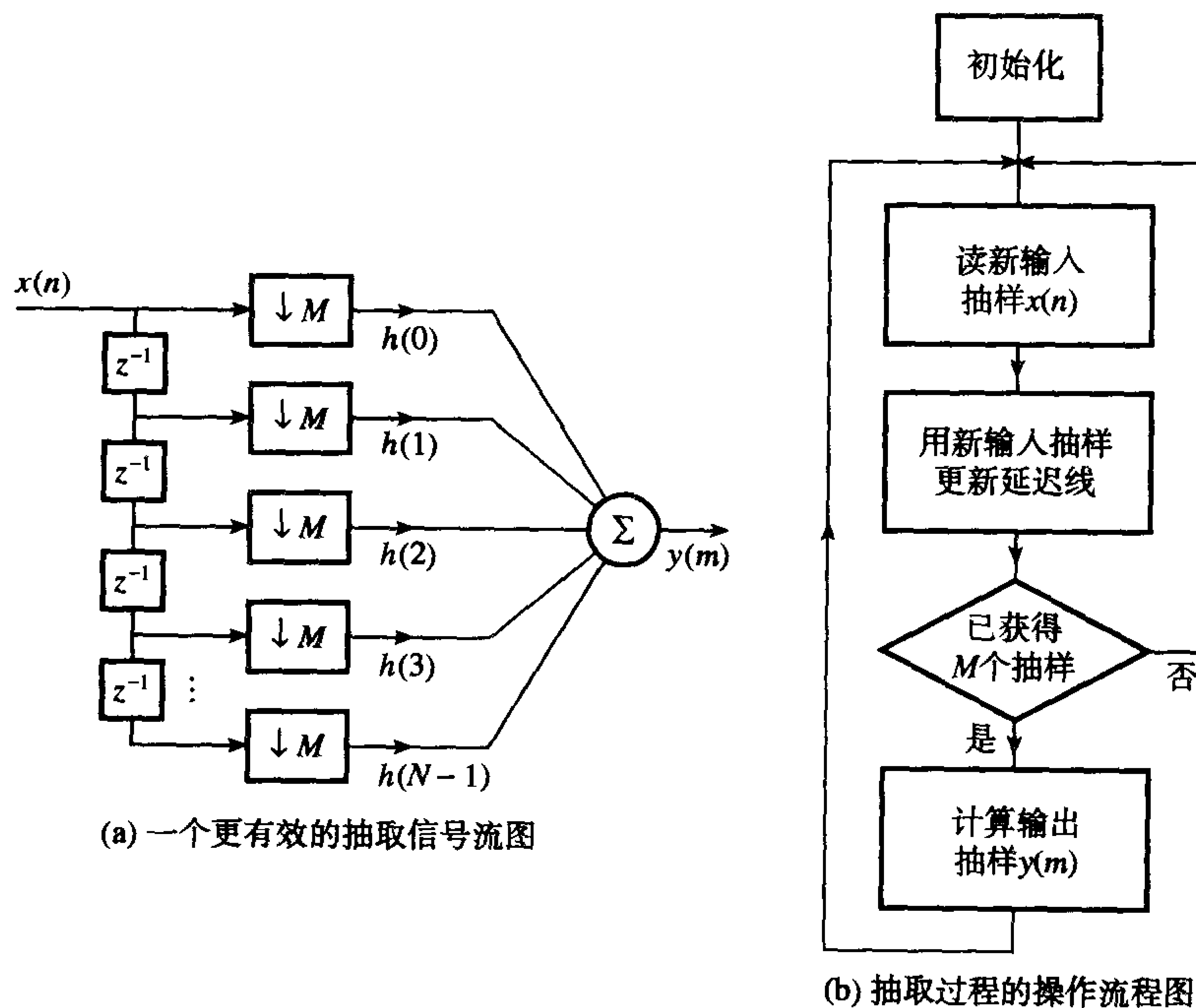
样，再保留下一个抽样，再抛弃接下来的 $M-1$ 个抽样；如此反复。由于对每个保留的抽样，其接下去的 $M-1$ 个 $w(n)$ 抽样被抛弃，所以不必要对这些抛弃的抽样也应用9.12a式。这样，抽取（抽样抛弃）操作可以在输入抽样与滤波系数的乘法运算之前进行（参见图9.14(a)）。含有滤波系数的乘法和加法以更低的抽样频率 F_s/M 执行，可使计算量降为原来的 M 分之一。



(a) 抽取滤波器的一个简单方框图解

(b) 抽取滤波器的信号流图

图 9.13 抽取滤波器的框图解和信号流图



(a) 一个更有效的抽取信号流图

(b) 抽取过程的操作流程图

图 9.14 更有效的抽取信号流图和抽取过程操作流程图

单级抽取滤波器的操作流程图由图9.14(b)给出。三级抽取滤波器的操作流程图在图9.15中给出，它是单级操作的直接拓展。

9.4.1 多级抽取程序

一个基于以上算法的独立交互C语言程序在附录9A中给出，它运行于PC（个人计算机）之上。程序最多可以分三级抽取输入数据，参见图9.15的流程图。每级抽取需要一个整数抽取因子和代表一个线性相位FIR数字滤波器的一组 N 点滤波器系数。

输入数据从PC的一个数据文件中读出，抽取后的数据输出到一个用户指定的文件。假定一个三级抽取（参见图9.15），第一级每隔M_1个输入抽样，一个输出抽样被计算出。每隔M_2个第一级的输出抽样，得到第二级的一个输出抽样。最后，每隔M_3个第二级的输出抽样，得到第三级的一个输出抽样。这样，每隔M个$x(n)$的输入抽样结束一次抽取循环，且$M = M_1M_2M_3$，一个输出抽样将被计算并存储到输出文件。该过程将重复执行直到完成所有的输入抽样。

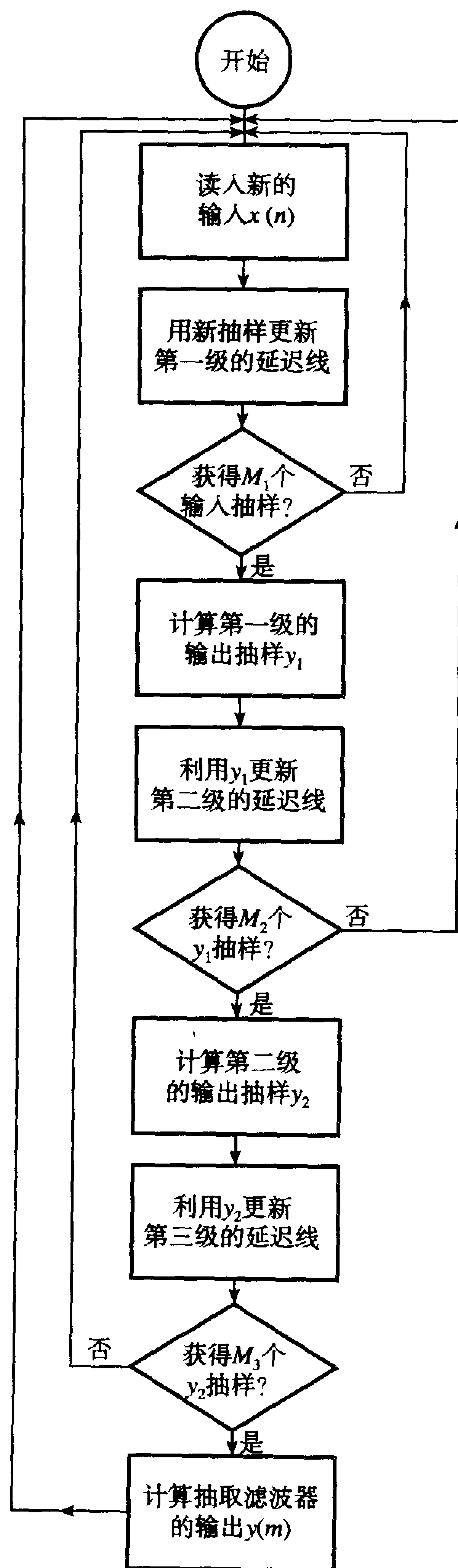


图9.15　一个三级抽取滤波器的流程图

该程序是完全独立的。使用这个程序，用户必须确定级数、总抽取因子和各级抽取因子、各级的直接结构的FIR线性滤波器系数。还有给出包含被抽取数据的文件名称、滤波器系数文件及存放输出数据的文件名称。

9.4.2 抽取过程的检验示例

用来检验抽取过程的输入序列是一个全通信号（Crochiere and Rabiner, 1979）

$$x(n) = -\alpha, \quad n = 0$$
$$= (1-\alpha^2)\alpha^{n-1}, \quad n = 1, 2, \ldots \tag{9.13}$$

其中 $\alpha = 0.9$。序列的前29个数据抽样在表9.4中列出。

在这个例子中，进行了一个二级抽取。抽取因子5和2分别用于第一和第二级，总的抽取因子为10。表9.4分别列出了个数为25和28的FIR滤波器系数，抽取结果也同时给出。

表9.4 抽取过程检验示例的数据

n	$x(n)$	$y(m)$	$h_1(k)$	$h_2(k)$
0	−0.9000		−0.000 174	−0.000 303
1	0.1900		−0.002 682	0.001 807
2	0.1710		−0.006 346	0.003 120
3	0.1539		−0.011 033	−0.001 169
4	0.1385		−0.014 156	−0.009 267
5	0.1247		−0.012 024	−0.007 792
6	0.1122		−0.000 775	0.011 124
7	0.1010		0.021 904	0.027 651
8	0.0909		0.055 181	0.007 674
9	0.0818	0.000 040	0.094 397	−0.045 444
10	0.0736		0.131 836	−0.064 816
11	0.0662		0.158 866	0.022 946
12	0.0596		0.168 728	0.202 371
13	0.0537		0.158 866	0.352 610
14	0.0483		0.131 836	0.352 610
15	0.0435		0.094 397	0.202 371
16	0.0391		0.055 181	0.022 946
17	0.0352		0.021 904	−0.064 816
18	0.0317		−0.000 775	−0.045 444
19	0.0285	−0.000 286	−0.012 024	0.007 674
20	0.0257		−0.014 156	0.027 651
21	0.0231		−0.011 033	0.011 124
22	0.0208		−0.006 346	−0.007 792
23	0.0187		−0.002 682	−0.009 267
24	0.0168		−0.000 174	−0.001 169
25	0.0152			0.003 120
26	0.0136			0.001 807
27	0.0123			−0.000 303
28	0.0110			
29	0.0099	0.001 116		
30	0.0089			
31	0.0081			
32	0.0072			
33	0.0065			
34	0.0059			
35	0.0053			
36	0.0048			
37	0.0043			
38	0.0039			
39	0.0035	−0.001 659		

（续表）

n	$x(n)$	$y(m)$	$h_1(k)$	$h_2(k)$
40	0.0031			
41	0.0028			
42	0.0025			
43	0.0023			
44	0.0020			
45	0.0018			
46	0.0017			
47	0.0015			
48	0.0013			
49	0.0012	−0.000 402		
50	0.0011			

$x(n)$和$y(m)$是输入和抽取后的数据。$h_1(k)$和$h_2(k)$是抽取滤波器的系数。

9.4.2.1　输出延迟

一个抽取滤波器的输出会延迟一定数量的抽样时间，取决于在抽取滤波中所使用滤波器的类型。假定滤波器是线性相位FIR滤波器，单级、双级和三级抽取的群延迟分别为

$$T（第一级）= \frac{1}{M}[T_1 - (M-1)] \text{ 抽样时间} \tag{9.14a}$$

$$T（第二级）= \frac{1}{M_1M_2}[T_1 + M_1T_2 - (M_1M_2-1)] \text{ 抽样时间} \tag{9.14b}$$

$$T（第三级）= \frac{1}{M_1M_2M_3}[T_1 + M_1T_2 + M_1M_2T_3 - (M_1M_2M_3-1)] \text{ 抽样时间} \tag{9.14c}$$

其中T_i是第i级滤波器延迟，由下式：

$$T_i = (N_i+1)/2 \text{ 抽样时间}$$

N_i是第i级滤波器系数的个数。在上面的二级检验示例中，滤波器分别延迟14.5和13个抽样时间，总的群延迟为

$$T(2\text{级}) = [1/(5\times 2)][14.5+5\times 13-(5\times 2-1)] = 7.05 \text{抽样时间}$$

如果希望得到一个整数延迟，那么应该选择N_i使得上面方程中的T是个整数，这种情形下输入和输出是可比的。例如在多抽样率高通滤波中输出抽样需要根据抽取滤波器或内插滤波器的延迟来做修正。

9.5　内插滤波器的软件实现

一个内插滤波器的框图表示在图9.16(a)中给出，信号流图则在图9.16(b)中。对每一个加到内插滤波器的输入抽样$x(n)$，内插器（带一个向上箭头的方框）将L-1个零值抽样插入到输入抽样$x(n)$之后，再将它们滤波得到$y(m)$。最终对$x(n)$的每一个输入抽样，获得L个$y(m)$的输出抽样。通过内插滤波器，输入抽样频率被有效地从F_s增加到LF_s。在每一个输入抽样后增加L-1个零表明每个输入信号（数据）的能量被分配到L个输出抽样中，即内插滤波器具有1/L的增益。在内插过程之后，应该将输出抽样乘以L来恢复信号（数据）能量到正常水平。

内插方程为

$$y(m) = \sum_{k=0}^{N-1} h(k)w(m-k) \tag{9.15a}$$

$$w(m-k) = \begin{cases} x[(m-k)/L], & m-k = 0, L, 2L, \ldots \\ 0 \end{cases} \tag{9.15b}$$

图 9.17 给出了一个简单内插过程的示意图，其中 $L=3$，滤波器长度为 10。一个输入抽样后跟两个零值加到延迟线，然后下一个输入抽样后又跟两个零值，反复如此。对每一个加到延迟线的抽样(数据或者零)计算出一个输出抽样。$x(n)$的两个输入抽样加到内插滤波器后延迟线的内容在图中显示出来。我们看到，对于每加入一个输入抽样，将计算出三个抽样。在延迟线中的非零抽样（$x(n)$中的实际抽样）被 $L-1$ 个零值（图例中是两个）所分隔。显然，与零值的乘法运算是不必要的。

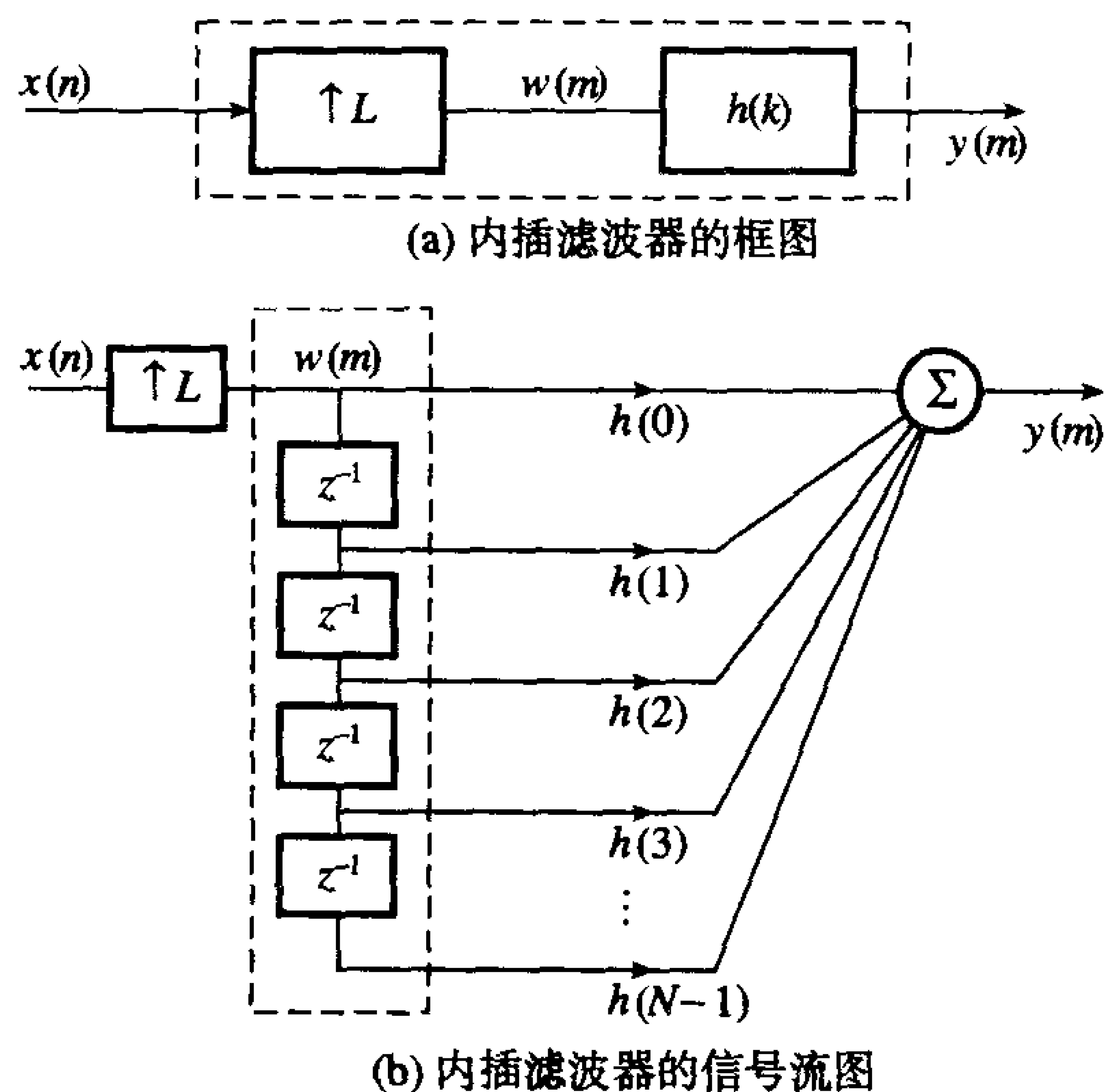


图 9.16　内插滤波器的框图和信号流图

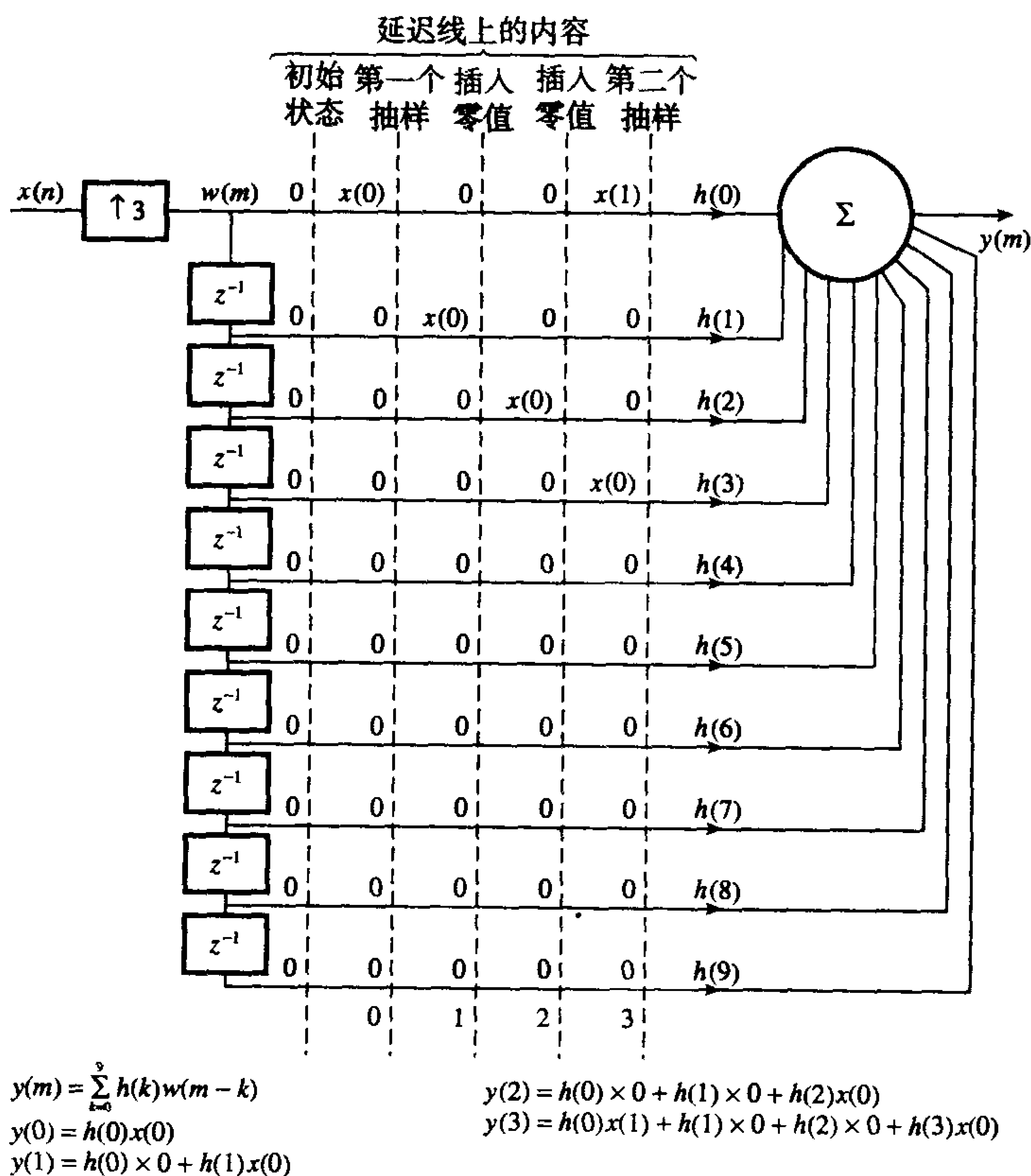


图 9.17　一个简单内插过程 $L=3$ 的示意图

图 9.18 给出了单级内插滤波器的操作流程图。在执行过程中，只有非零值用于参与输出抽样的计算。最高为三级的内插滤波器的操作流程图在图 9.19 中给出。

另一种有效的实现方法，称为多相滤波（Crochiere and Rabiner, 1983），该方法充分利用了在延迟线上一些值为零的特性。它完全取消了内插器，因此不必要储存那些零值抽样。延迟线被缩减到原先的 N/L 长度。在这种方法中，对于每个加到延迟线的输入抽样，N/L 个延迟线抽样参与计算 L 个输出抽样，且每个输出抽样使用了不同的一组滤波器系数（也就是说，那些对应着零值抽样的滤波器系数被省去了）。多相滤波法的局限是 N/L 的比值必须是一个整数。在下一章，我们将介绍一个三级内插滤波器的 C 语言实现程序。

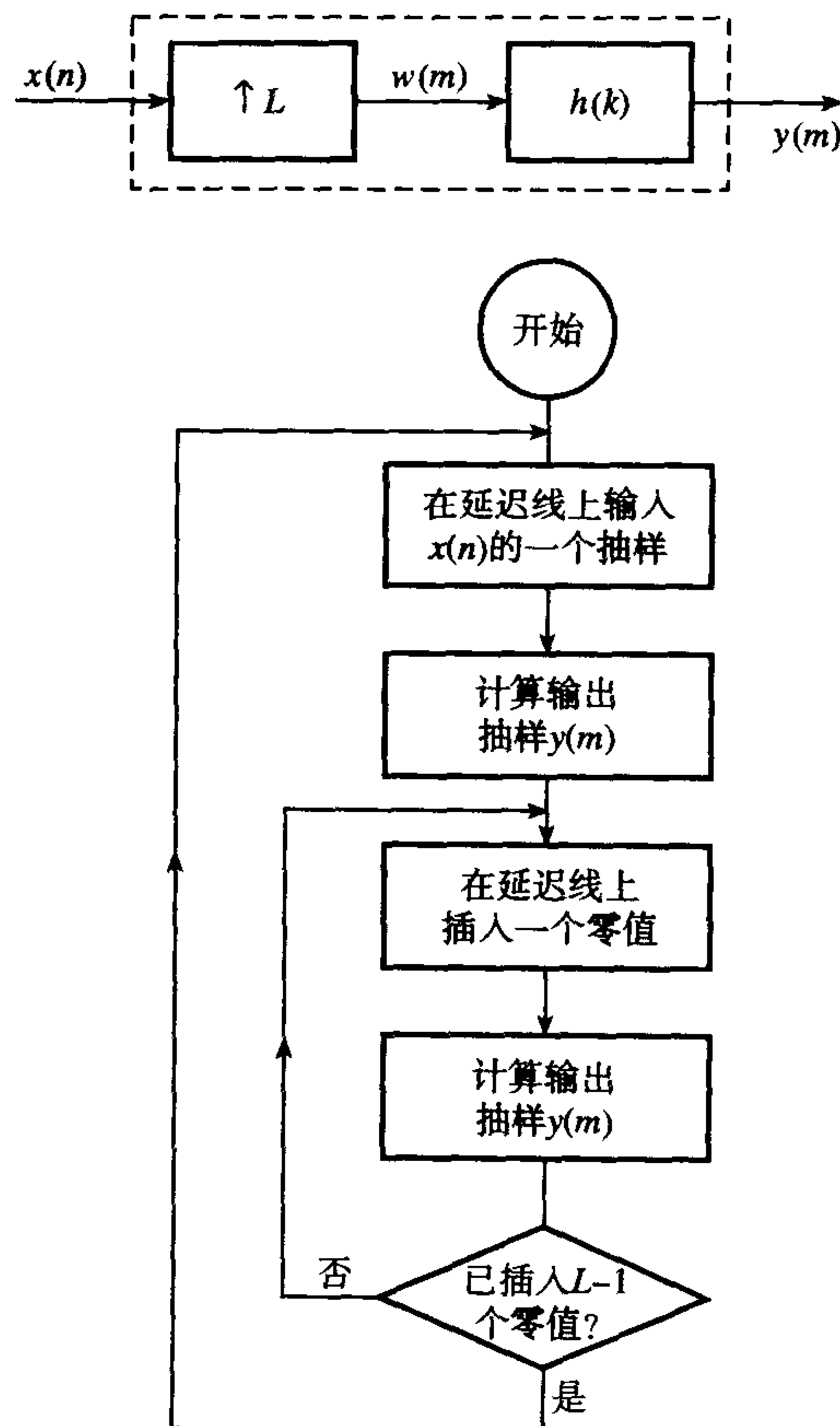


图 9.18　内插过程的流程图

9.5.1　多级内插程序

程序使用了上一节的计算方法。该程序能对输入数据最高进行三级内插，参见图 9.19 的流程图。注意到由于抽取和内插的对偶性（一个抽取滤波器与一个内插滤波器组成了一个抽样率变换对），内插过程是标号反转的抽取过程。每级内插需要一个整数内插因子和代表一个线性相位 FIR 数字滤波器的一组 N 点滤波器系数。程序运行于一台个人计算机上，但稍做修改也能运行于其他机器上。

输入数据从计算机上的一个数据文件中读出，内插后的数据写到一个用户指定的输出文件。假定一个三级内插（参见图 9.19），在内插滤波器的第三级计算 L_3 个抽样。对于第三级的每个抽样，第二级相应计算 L_2 个抽样。对于第二级获得的每个抽样，第 1 级相应计算 L_1 个输出抽样。在循环

的结束，对应每个输入抽样 $x(n)$，L 个输出抽样（其中 $L=L_1L_2L_3$）被计算和存储到输出数据文件。该过程将重复执行直到所有的输入抽样都被处理。在本书指导手册的 CD 中有多级内插的源程序（详见前言）。

该程序是独立的。使用这个程序，用户必须给定级数、总内插因子、各级内插因子和各级 FIR 滤波器系数组。还要给定包含输入数据、滤波器系数及存放输出结果的文件名。

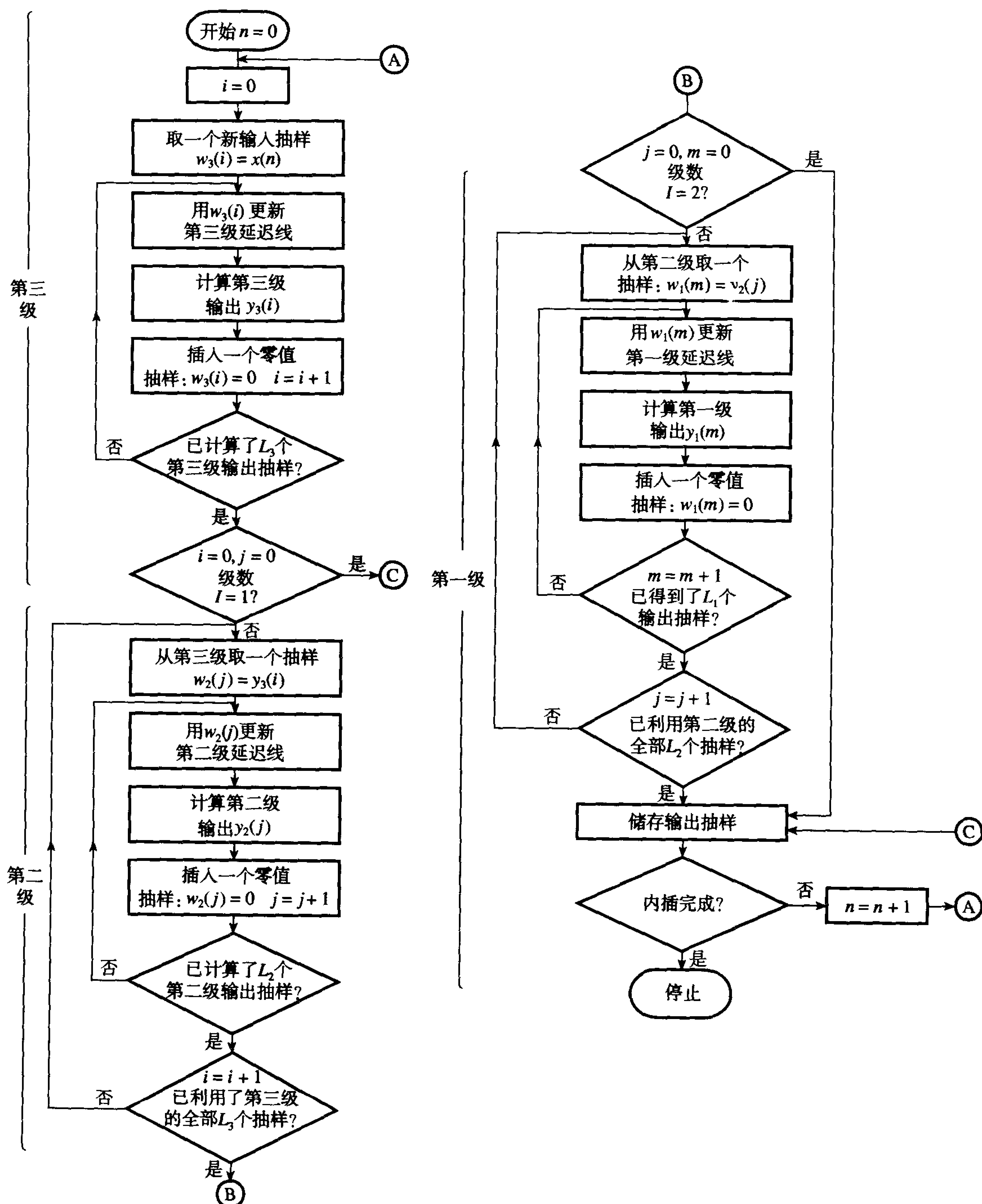


图 9.19 最高为三级的内插过程。例如 $I=1$，则只有标明第三级的部分被执行

9.5.2 检验示例

所使用的输入序列与抽取滤波器中的完全一致。表 9.5 给出了序列的前 5 个抽样。

表 9.5　内插过程检验示例的数据

n	$x(n)$	$y(m)$	$h_2(k)$	$h_1(k)$
0	−0.9000	$-4.744\,98\times10^{-7}$	−0.000 303	−0.000 174
		$-7.313\,815\times10^{-6}$	0.001 807	−0.002 682
		$-1.730\,554\times10^{-5}$	0.003 120	−0.006 346
		$-3.008\,7\times10^{-5}$	−0.001 169	−0.011 033
		$-3.860\,341\times10^{-5}$	−0.009 267	−0.014 156
		$-2.995\,969\times10^{-5}$	−0.007 792	−0.012 024
		$4.150\,394\times10^{-5}$	0.011 124	−0.000 775
		$1.629\,372\times10^{-4}$	0.027 651	0.021 904
		$3.299\,083\times10^{-4}$	0.007 674	0.055 181
		$4.876\,397\times10^{-4}$	−0.045 444	0.094 397
1	0.1900	$5.600\,492\times10^{-4}$	−0.064 816	0.131 836
		$5.226\,861\times10^{-4}$	0.022 946	0.158 866
		$2.857\,456\times10^{-4}$	0.202 371	0.168 728
		$-1.480\,226\times10^{-4}$	0.352 610	0.158 866
		$-7.700\,115\times10^{-4}$	0.352 610	0.131 836
		−0.001 544 5	0.202 371	0.094 397
		$-2.448\,377\times10^{-3}$	0.022 946	0.055 181
		$-3.400\,52\times10^{-3}$	−0.064 816	0.021 904
		$-4.320\,959\times10^{-3}$	−0.045 444	−0.000 775
		$-5.079\,387\times10^{-3}$	0.007 674	−0.012 024
2	0.1710	$-5.534\,875\times10^{-3}$	0.027 651	−0.014 156
		$-5.728\,923\times10^{-3}$	0.011 124	−0.011 033
		$-5.466\,501\times10^{-3}$	−0.007 792	−0.006 346
		$-4.756\,987\times10^{-3}$	−0.009 267	−0.002 682
		$-3.522\,772\times10^{-3}$	−0.001 169	−0.000 174
		$-1.715\,353\times10^{-3}$	0.003 120	
		$5.557\,998\times10^{-4}$	0.001 807	
		$3.324\,822\times10^{-3}$	−0.000 303	
		$6.400\,164\times10^{-3}$		
		$9.565\,701\times10^{-3}$		
3	0.1539	0.012 597 6		
		$1.544\,317\times10^{-2}$		
		$1.774\,401\times10^{-2}$		
		$1.933\,819\times10^{-2}$		
		$1.984\,539\times10^{-2}$		
		$1.894\,878\times10^{-2}$		
		$1.681\,832\times10^{-2}$		
		$1.303\,225\times10^{-2}$		
		$7.845\,689\times10^{-3}$		
		$1.357\,867\times10^{-3}$		
4	0.1385	$-6.262\,392\times10^{-3}$		
		$-1.462\,789\times10^{-2}$		
		$-2.343\,143\times10^{-2}$		
		$-3.207\,272\times10^{-2}$		
		$-3.972\,186\times10^{-2}$		
		$-4.567\,938\times10^{-2}$		
		$-4.993\,166\times10^{-2}$		
		$-5.142\,782\times10^{-2}$		
		$-5.009\,625\times10^{-2}$		
		$-4.527\,419\times10^{-2}$		

$x(n)$和$y(m)$是输入和抽取后的数据。$h_1(k)$和$h_2(k)$是抽取滤波器的系数。

在这个例子中，进行了二级内插过程。内插因子2和5分别用于第三级和第二级，使总的内插因子为10。表9.5列出了内插的结果，还有所使用的FIR滤波器系数，长度分别是25和28。

9.5.2.1 输出延迟

内插滤波器的输出比输入会延迟一定数量的抽样时间。单级、双级和三级内插的群延迟分别为

$$T\text{（第一级）} = T_1 \text{抽样时间} \tag{9.16a}$$

$$T\text{（第二级）} = T_1 + M_1T_2 \text{抽样时间} \tag{9.16b}$$

$$T\text{（第三级）} = T_1 + M_1T_2 + M_1M_2T_3 \text{抽样时间} \tag{9.16c}$$

其中T_i是第i级滤波器延迟：$T_i = (N_i+1)/2$抽样时间。N_i是第i级滤波器系数的个数。在上面的检验示例中，滤波器分别延迟13和14.5个抽样时间，总的群延迟为$13 + 2 \times 14.5 = 42$个抽样时间。

如果希望得到一个整数延迟，那么应该选择N_i使得9.16式能最终得到整数的总群延迟。如果内插滤波器的输入和输出是可比的（例如，在高通窄带滤波中，滤波运算是作为低通的逆过程执行的），那么输出信号的延迟可以被修正或调整。

9.6 利用多相滤波器结构实现抽样率变换

另一种能有效实现抽取或内插滤波器的方法是称为多相滤波器的结构。为了简便起见，我们首先考虑内插滤波器的多相滤波。

9.6.1 内插滤波器的多相实现

内插滤波器的多相实现巧妙利用了内插滤波器延迟线上的某些抽样是零值的特点。在这种情况下，内插器被去除，从而无需存储零值抽样。延迟线将减短到原长度的N/L（其中N是去镜像滤波器的长度，L是内插因子）。在这种方法里，对加到延迟线的每个输入抽样，N/L个延迟线上抽样参用以计算L个输出抽样，但每个输出抽样使用的是不同的滤波器系数组（即那些对应零值抽样的滤波器系数被省去了）。

为了更深刻地理解内插滤波器的多相实现，我们检验一个简单的1 ↑ 3内插滤波器，请参见图9.20。在这个例子中，$L=3$，滤波器系数个数为$N=8$。

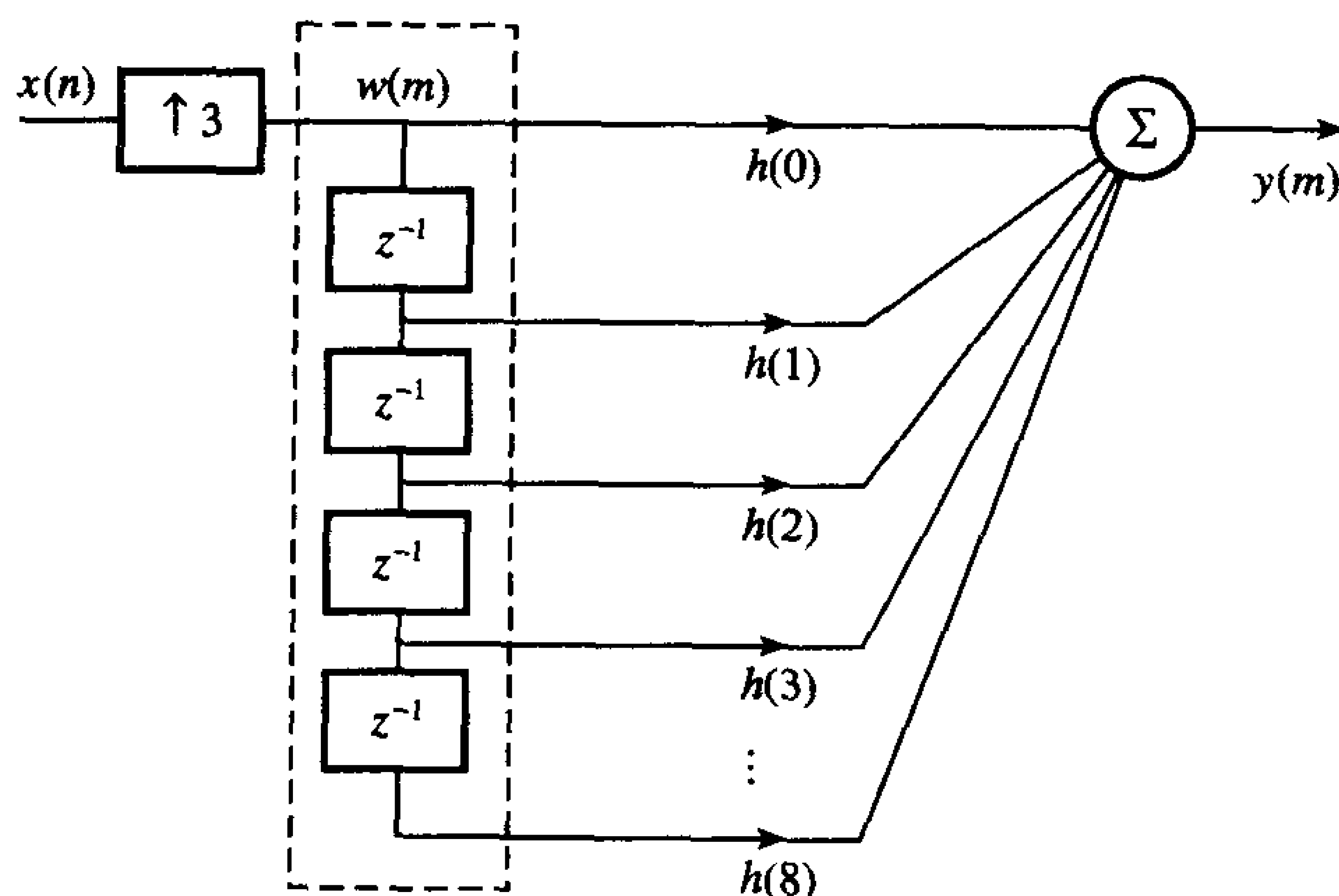
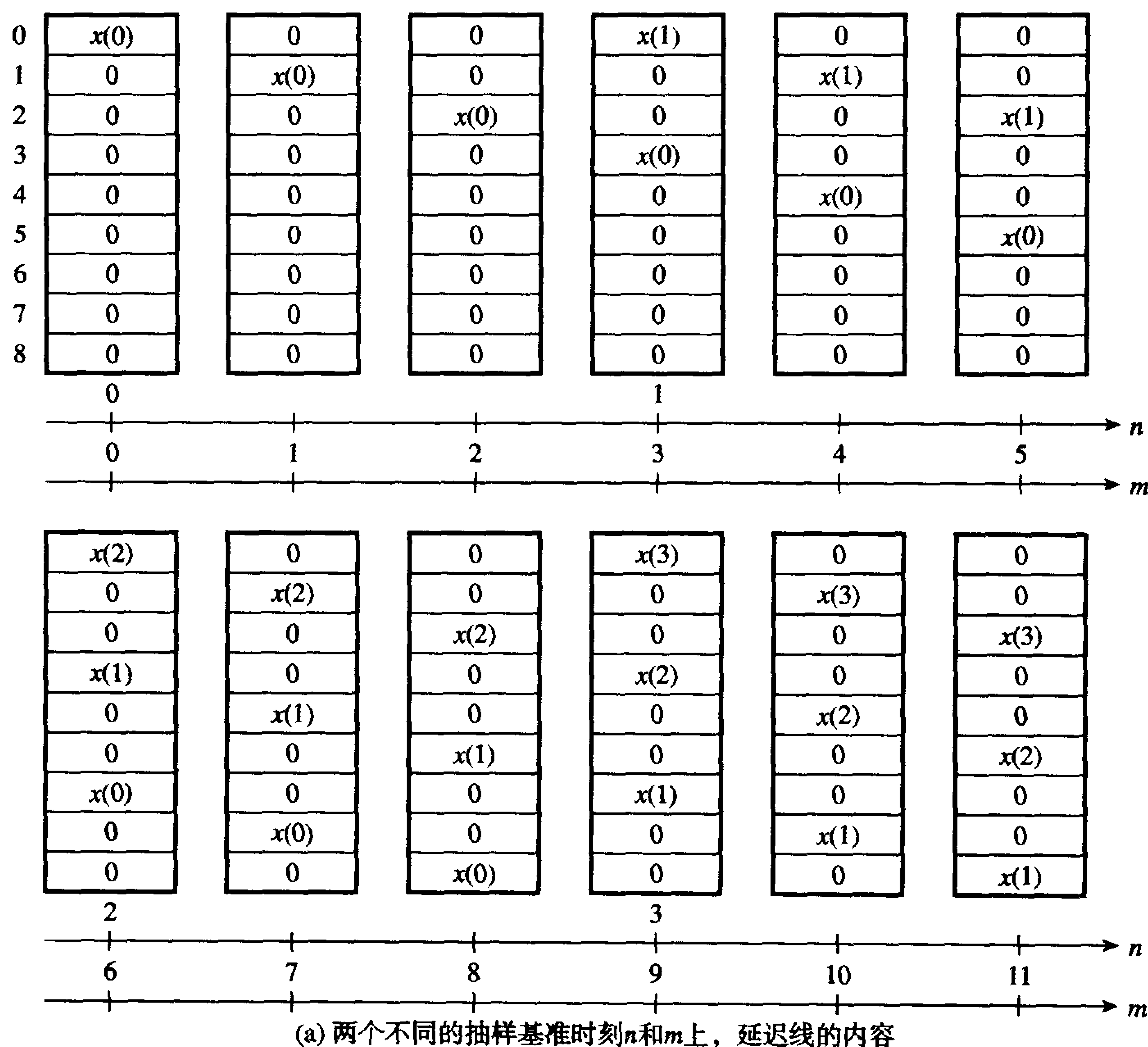


图9.20 一个1 ↑ 3内插滤波器，使用9点直接形式的FIR滤波器

参考图9.20，延迟线加入一个输入抽样后跟两个$(L-1)$零值，再接一个输入抽样后跟两个零值，如此重复。

加入四个输入抽样——$x(0)$、$x(1)$、$x(2)$、$x(3)$及相应的零值后，延迟线上的内容和对应的两种抽样率的抽样时刻如图 9.21 所示。



(a) 两个不同的抽样基准时刻n和m上，延迟线的内容

$$y(0) = x(0)h(0)$$
$$y(1) = x(0)h(1)$$
$$y(2) = x(0)h(2)$$

$$y(3) = x(1)h(0) + x(0)h(3)$$
$$y(4) = x(1)h(1) + x(0)h(4)$$
$$y(5) = x(1)h(2) + x(0)h(5)$$

$$y(6) = x(2)h(0) + x(1)h(3) + x(0)h(6)$$
$$y(7) = x(2)h(1) + x(1)h(4) + x(0)h(7)$$
$$y(8) = x(2)h(2) + x(1)h(5) + x(0)h(8)$$

$$y(9) = x(3)h(0) + x(2)h(3) + x(1)h(6)$$
$$y(10) = x(3)h(1) + x(2)h(4) + x(1)h(7)$$
$$y(11) = x(3)h(2) + x(2)h(5) + x(1)h(8)$$

(b) 多相滤波器的输出抽样

图9.21　一个使用多相结构的 1 ↑ 3 内插过程

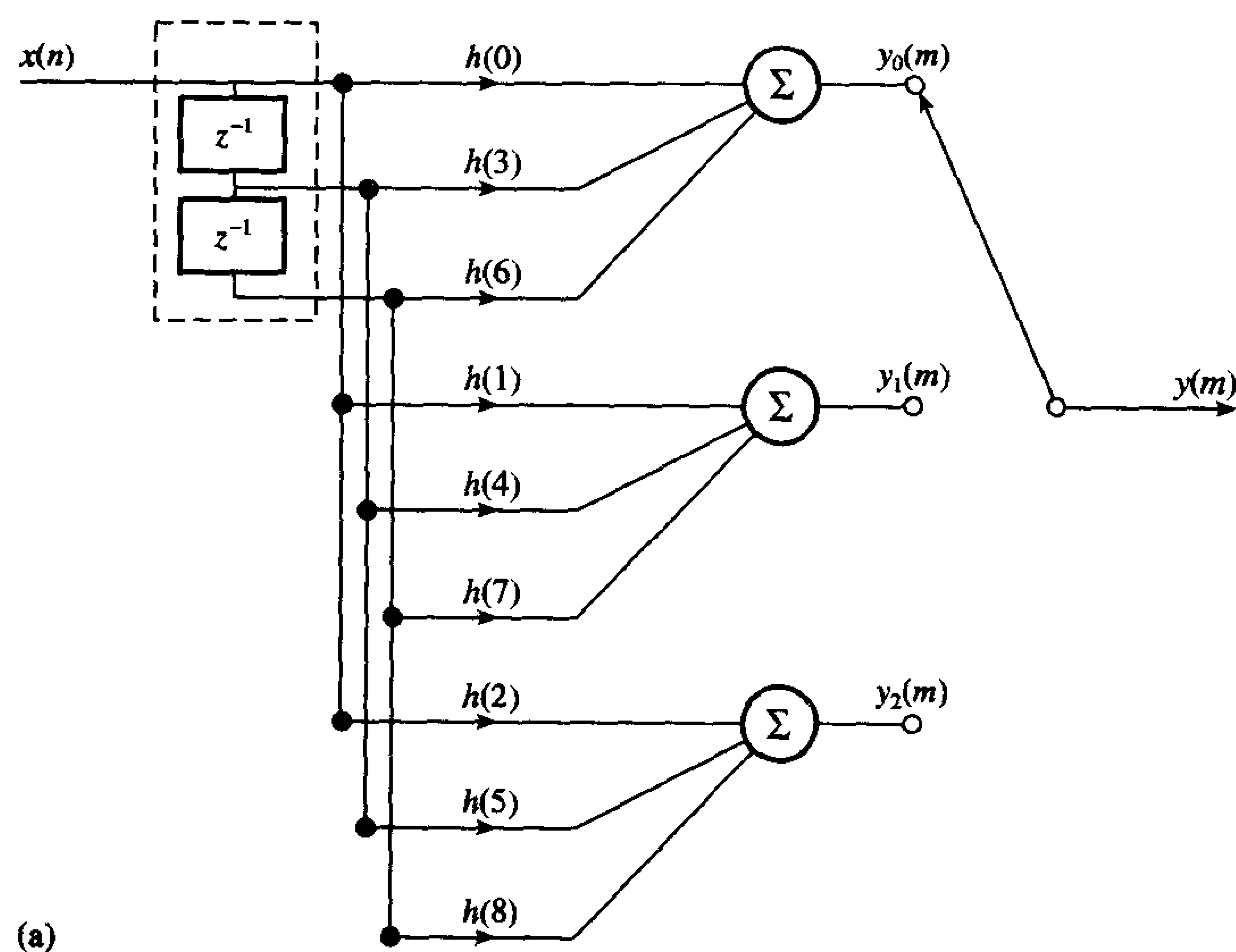
对每个加到延迟线的抽样（输入数据或零值），计算出一个输出抽样。因此，对每个实际加入的抽样，将计算出三个输出抽样。非零抽样（如延迟线上的实际抽样$x(n)$）被$L-1$个（本例是两个）零值分隔。显然，与零值抽样的乘法运算是不必要的。

内插滤波器的连续输出抽样在图 9.21 中给出。在这种实现结构中，只有输入的非零值抽样参与产生输出抽样的计算。下面给出应注意的要点：

(1) 每个输入抽样产生三个输出抽样，这些输出抽样使用三个子滤波器获得，且三组滤波器为

- {h(0), h(3), h(6)}, {h(1), h(4), h(7)}, {h(2), h(5), h(8)}；
- 对每个新的输入抽样，该过程被重复执行；
- 子滤波器工作于较低的抽样率。

(2) 总的滤波器由数个子滤波器的并行运算而有效地实现：参见图9.22(a)。子滤波器，或称为多相滤波器，共同分享一个延迟线，使总的存储需求按因子3降低。对于每个新的输入抽样，每个多相滤波器在较低的抽样率上各提供一个输出抽样：参见图9.22(b)。



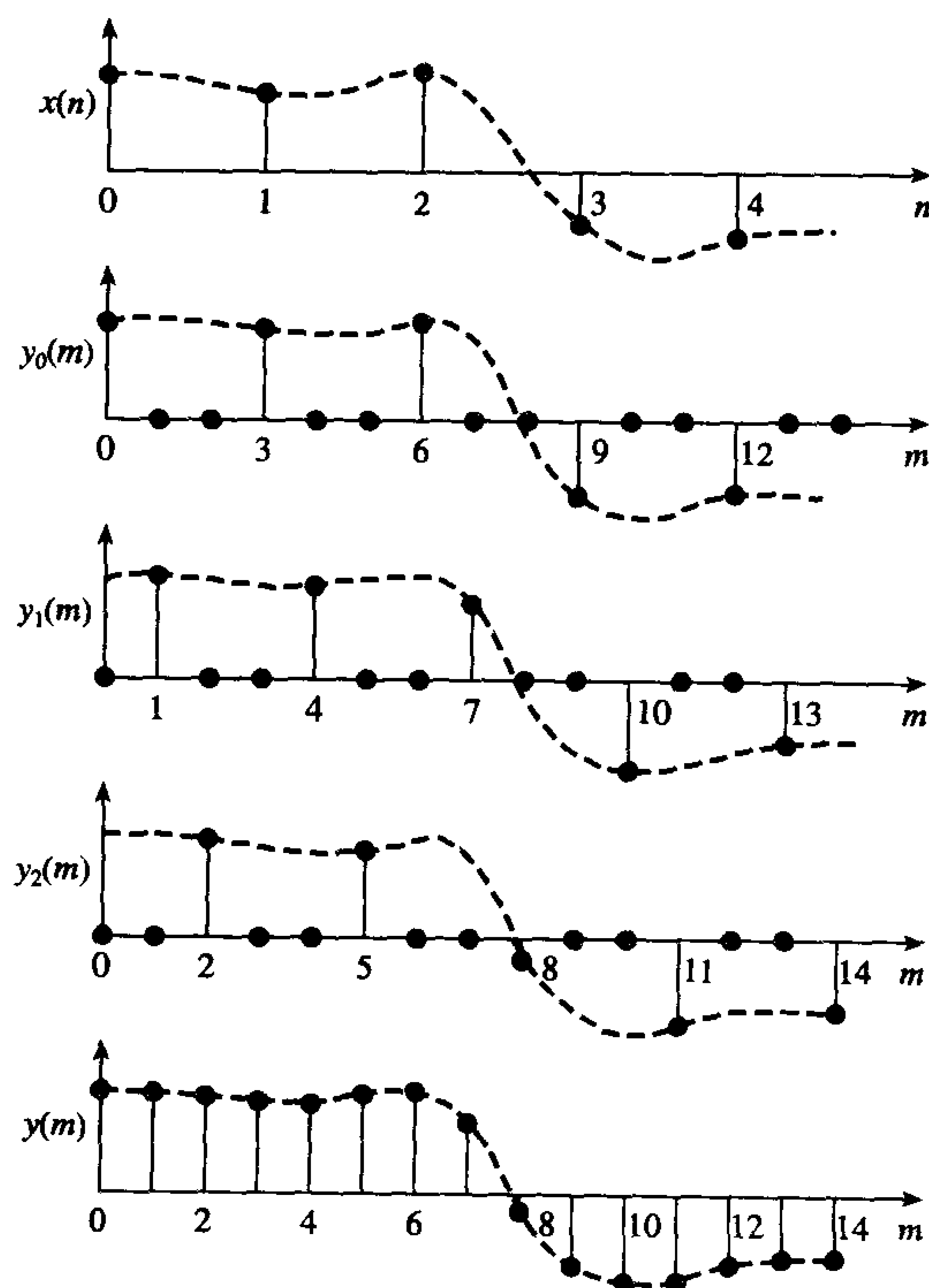
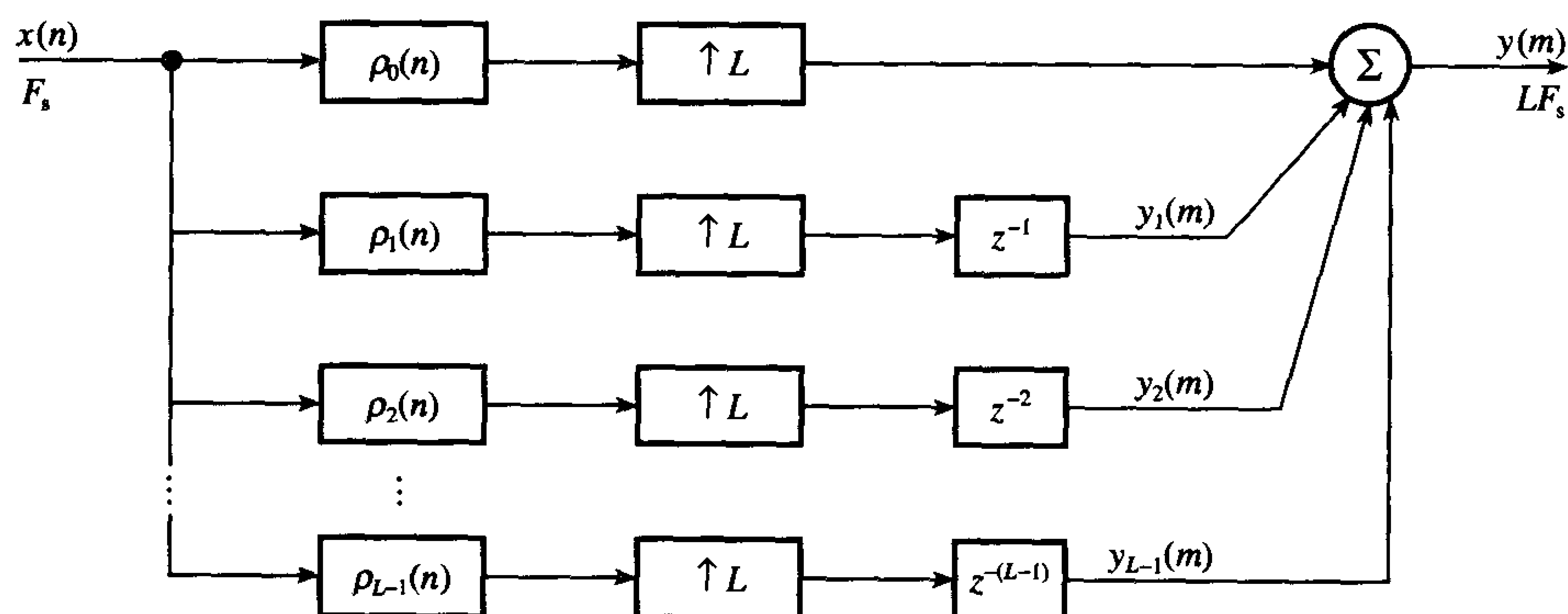


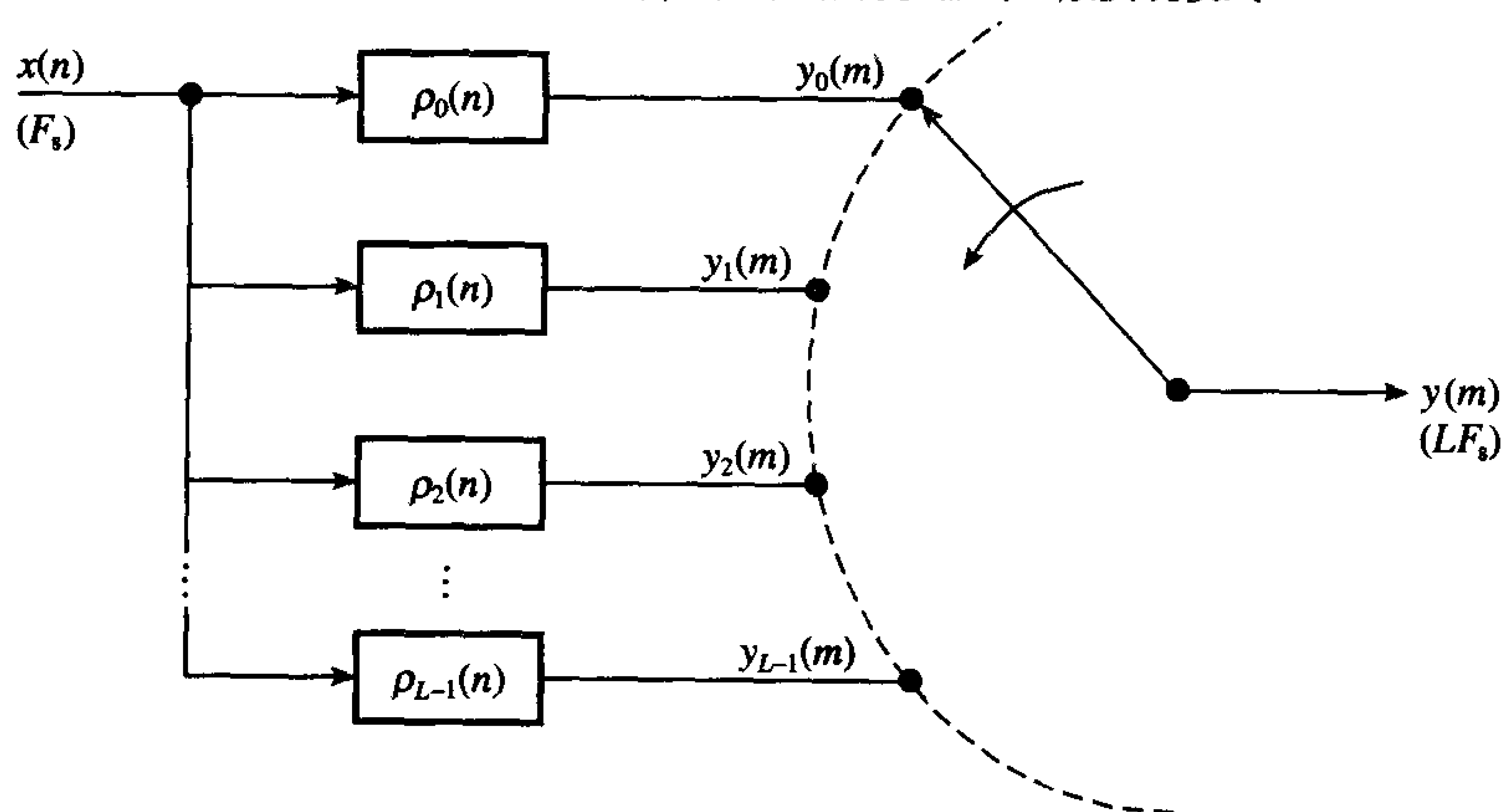
图9.22　多相滤波器实现1↑3内插

内插滤波器的多相滤波实现的一般结构请参见图9.23。在该图中，对$x(n)$的每个输入抽样有L个$y(m)$的输出抽样，每个多相滤波器各自产生一个。例如图上面的多相滤波器$\rho_0(n)$，产生输出抽样$y_0(n)$；下方的多相滤波器$\rho_1(n)$，则产生输出抽样$y_1(n)$；其余的多相滤波器也是如此。在实际中，多相滤波器通常利用图9.23(b)中的转接器模式来实现。转接器从起始的顶部位置按逆时针方向旋转。

多相滤波器的分析表明它们是全通滤波器，但通过它们的附加相移是不同的。这就是为什么这些滤波器称为多相滤波器的原因。多相滤波结构的有效性来源于将单个N点FIR滤波器分解成一组长度为N/L的子滤波器，其中N应该选定为L的整数倍，上述的滤波操作是在一个较低的抽样率上进行的。



(a) 内插多相滤波器的一般实现模式



(b) 内插多相滤波器的转接器实现模式

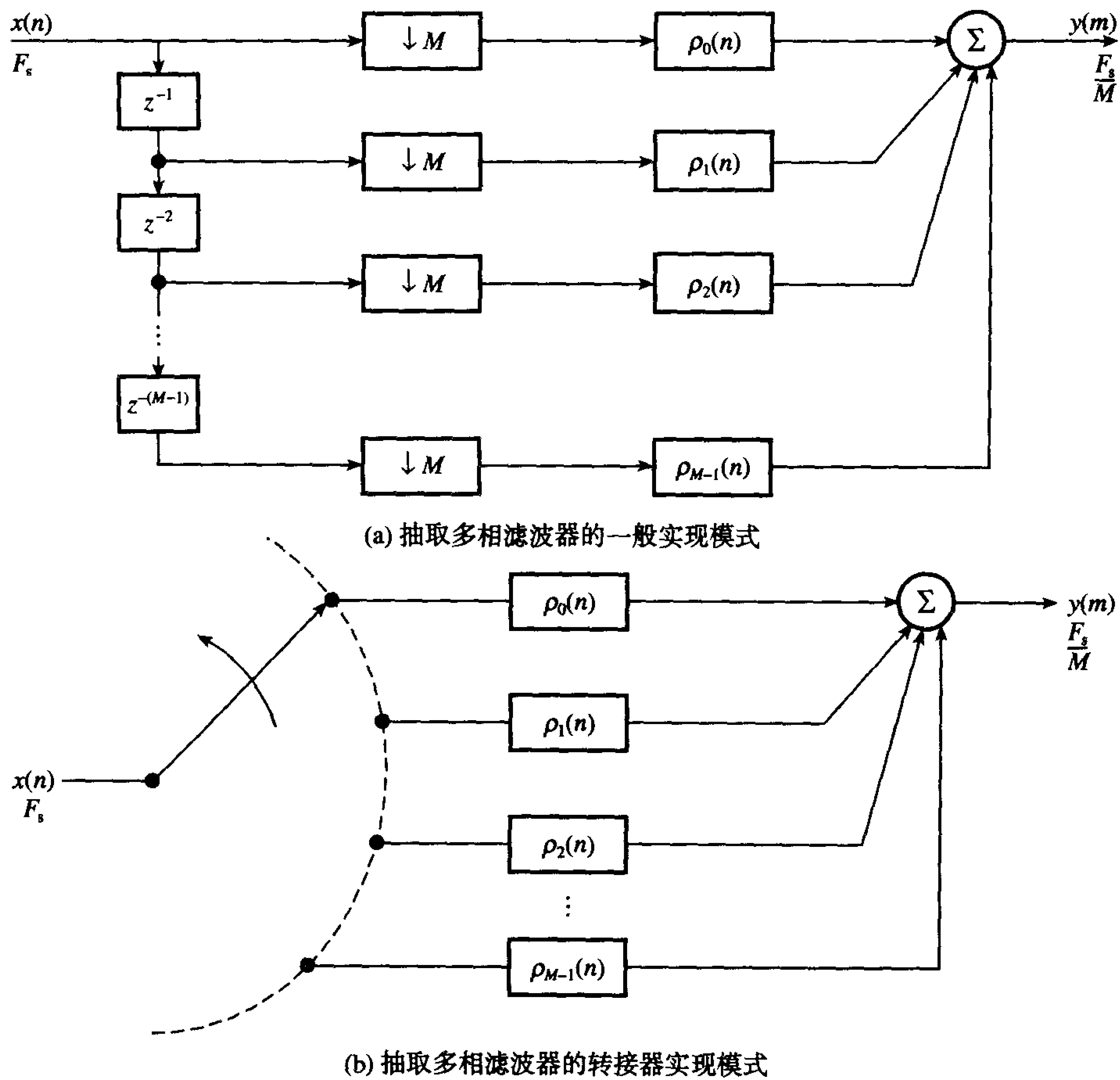
图 9.23　内插多相滤波器的实现模式

内插多相滤波器的系数由下式给出：

$$\rho_k(n) = h(k + nL), \quad k = 0, 1, \ldots, L-1; \quad n = 0, 1, \ldots, \frac{N}{L} - 1$$

抽取多相滤波器的实现可以从图9.23内插滤波器的系统转置来得到，结果见图9.24。注意到这次转接器从起始的顶部位置，即抽样时刻$m = 0$，按顺时针方向旋转。在这种情形下，多相滤波器与原先的抽取滤波器具有以下关系：

$$\rho_k(n) = h(k + nM), \quad k = 0, 1, \ldots, M-1; \quad n = 0, 1, \ldots, \frac{N}{M} - 1$$



(a) 抽取多相滤波器的一般实现模式

(b) 抽取多相滤波器的转接器实现模式

图 9.24 抽取多项滤波器的实现模式

9.7 应用举例

数字音频工程领域已经从多抽样率技术中获得了巨大的收益。例如，将它们应用于CD播放器以简化D/A转换过程，同时还能保证复制声音的质量。在数字音频系统的前端，努力的目标集中于δ调制技术，并结合多抽样率处理，用以从模拟音频信号中获得高质量数字数据。

其他应用多抽样率技术的领域还有高质量数据获取、高分辨率谱分析、窄带数字滤波的设计和实现等。

在下一节，我们将介绍一系列这些方面的应用。

9.7.1 数字音频的高质量模/数转换

在数字音频领域，不断增加的对高质量、高分辨率和高速ADC的需求导致了使用Σ-Δ调制技术的单比特ADC的出现。这为取消一个音频系统前端的（数/模）转换处理中大多数的模拟电路，包括抗混叠滤波器和抽样－保持电路等，提供了可能性。

图 9.25 给出了一个简化的快速单比特 ADC 处理的框图（Adams, 1986; Matsuya et al., 1987; Welland et al., 1989）。模拟音频信号首先利用Σ-Δ调制于3.072 MHz抽样率转换成一个单比特流。再利用一个多级抽取滤波器，降低单比特流的抽样率到48 kHz，产生一个16个比特的PCM（脉冲编码调制）码字。现在已有许多利用多抽样率技术实现的 ADC 的现货供应。例如 Crystal

Semiconductor生产的16和18位立体ADC（CS5326，CS5327，CS5328，CS5329），以及摩托罗拉生产的ADC（DSP56ADC16）等。

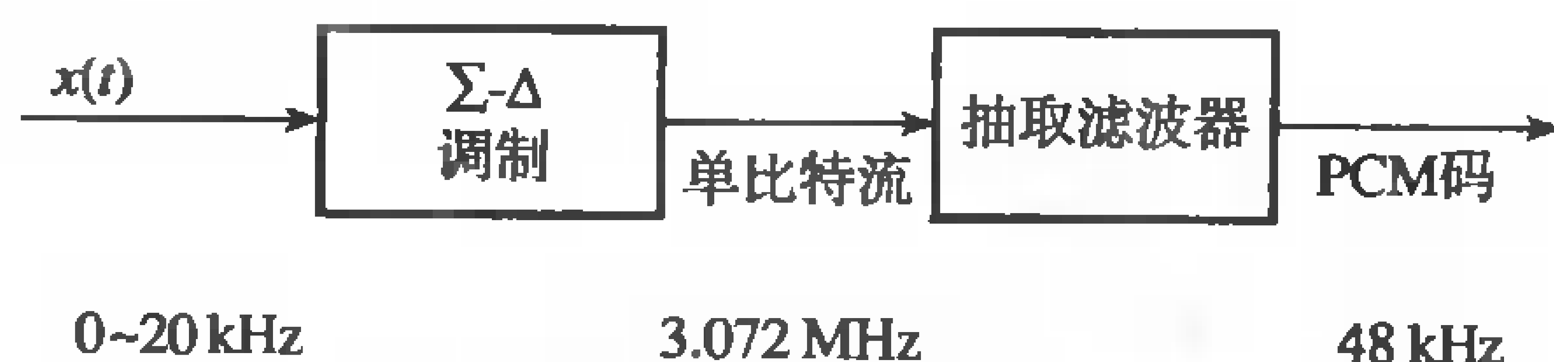


图9.25　简化的单比特ADC处理的框图

9.7.2　CD播放器高保真系统中的高效模/数转换

多抽样率技术的一个首要应用是在高保真CD播放器中再现声音和音乐。

图9.26给出了重建CD中模拟音频信号过程的图示。在解码和纠错后，数字信号的字长是16个比特，代表了44.1 kHz抽样率的声音信息。

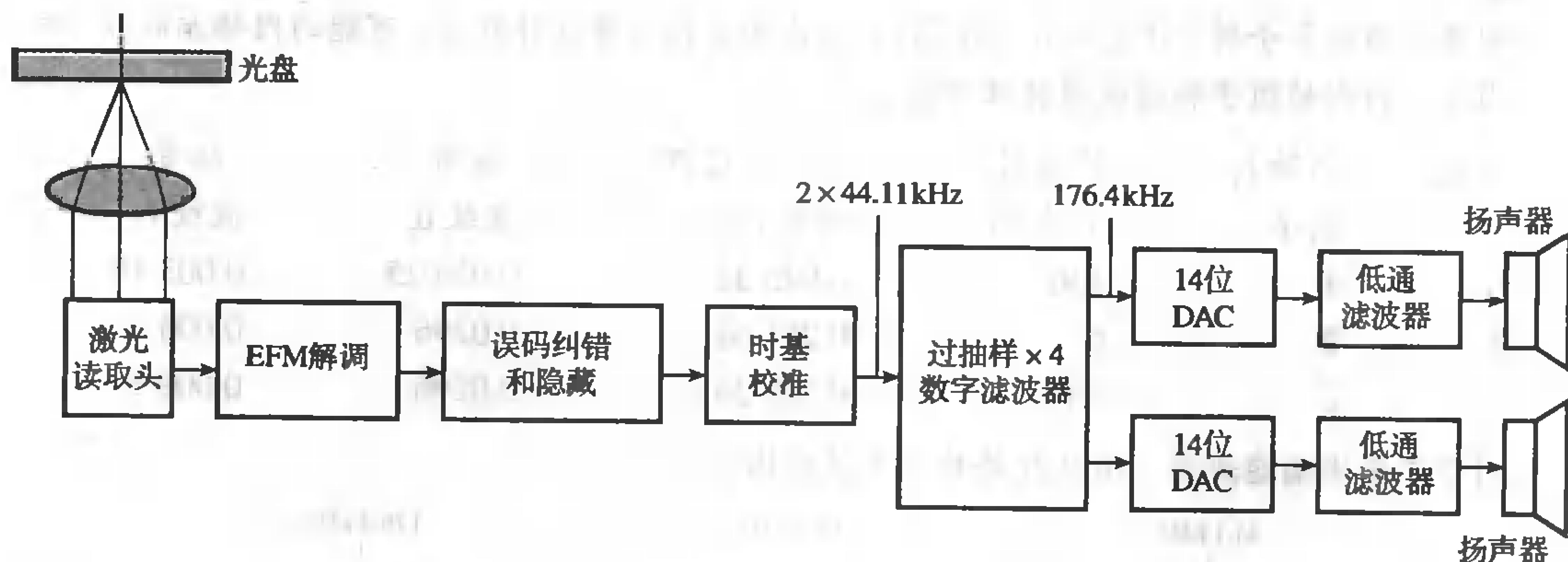


图9.26　CD播放器系统音频信号的重现

如果这些数字码直接转换成模拟信号，会产生以抽样频率44.1 kHz的倍数为中心的镜像频带。尽管镜像频率是听不见的，因为超出了基带的0～20 kHz。但如果将它们传给使用者的功放和扬声器，则会造成超负荷工作，也可能产生互调失真现象。因此基带以外的频率分量有必要衰减至少50 dB。很难做出能满足这种衰减的模拟滤波器，同时还需精调以保证两路立体声通道的匹配。

为了避免模拟滤波器问题，CD播放器采用了多抽样率滤波技术。即在DAC之前按因子4内插，提高抽样频率到176 kHz（4 × 44.1 kHz = 176.4 kHz）。在时域，这么做的结果是使信号具有更精细的量化步长。在频域，镜像频率现在被推到一个更高的频率上，使镜像频率更容易滤去。因此，在D/A转换后只需要一个相对简单的低通滤波器。在实际应用中，数字滤波器进行一个sin x/x校正（参见第2章）以补偿DAC后的保持电路的影响。sin x/x校正的好处在于能够将信号在174 kHz的两边均衰减超过18 dB，这进一步简化了对模拟去镜像滤波器的需求。在内插后使用一个简单的三阶贝塞尔（Bessel）滤波器来实现剩余的衰减需求，它的3 dB截止频率为30 kHz，在通带具有合理的线性相位响应。

将数据过抽样还有其他益处。如降低噪声门限，因为量化噪声现在被分散到一个更宽的频带上，使采用较低位数的DAC成为可能，仍能保证达到与16位DAC同样的信噪比。因此，在图9.26中，内插的16个比特的数据经过抽样和噪声整形后，在加到14位DAC之前被舍入成14位。

市场上还有其他利用本章的过抽样概念的DAC。例如Philips公司的单比特流DAC（SAA7322，SAA7323，SAA7350）。

例9.5 一个数字音频系统，利用过抽样技术来降低对模拟去镜像滤波器的需求。系统总的滤波器指标如下：

基带	0~20 kHz
输入抽样频率 F_s	44.1 kHz
输出抽样频率	176.4 kHz
阻带衰减	50 dB
通带波纹	0.5 dB
过渡带宽	2 kHz
阻带边沿频率	22.05 kHz

设计一个合适的内插滤波器。

解：

利用本书指导手册（详见前言）的CD中给出的多抽样率设计程序，可能的内插滤波器（整数因子）的内插因子和滤波器特性归结如下。

级数	内插 L_i 因子	滤波器长度 N_i	归一化过渡带宽 Δf_i	通带波纹 δ_p	阻带波纹 δ_s
1	4	146	0.045 35	0.059 25	0.003 16
2	2	6	0.261 62	0.0296	0.003 16
	2	83	0.273 24	0.0296	0.003 16

对于二级内插滤波器，图9.27给出了系统框图。

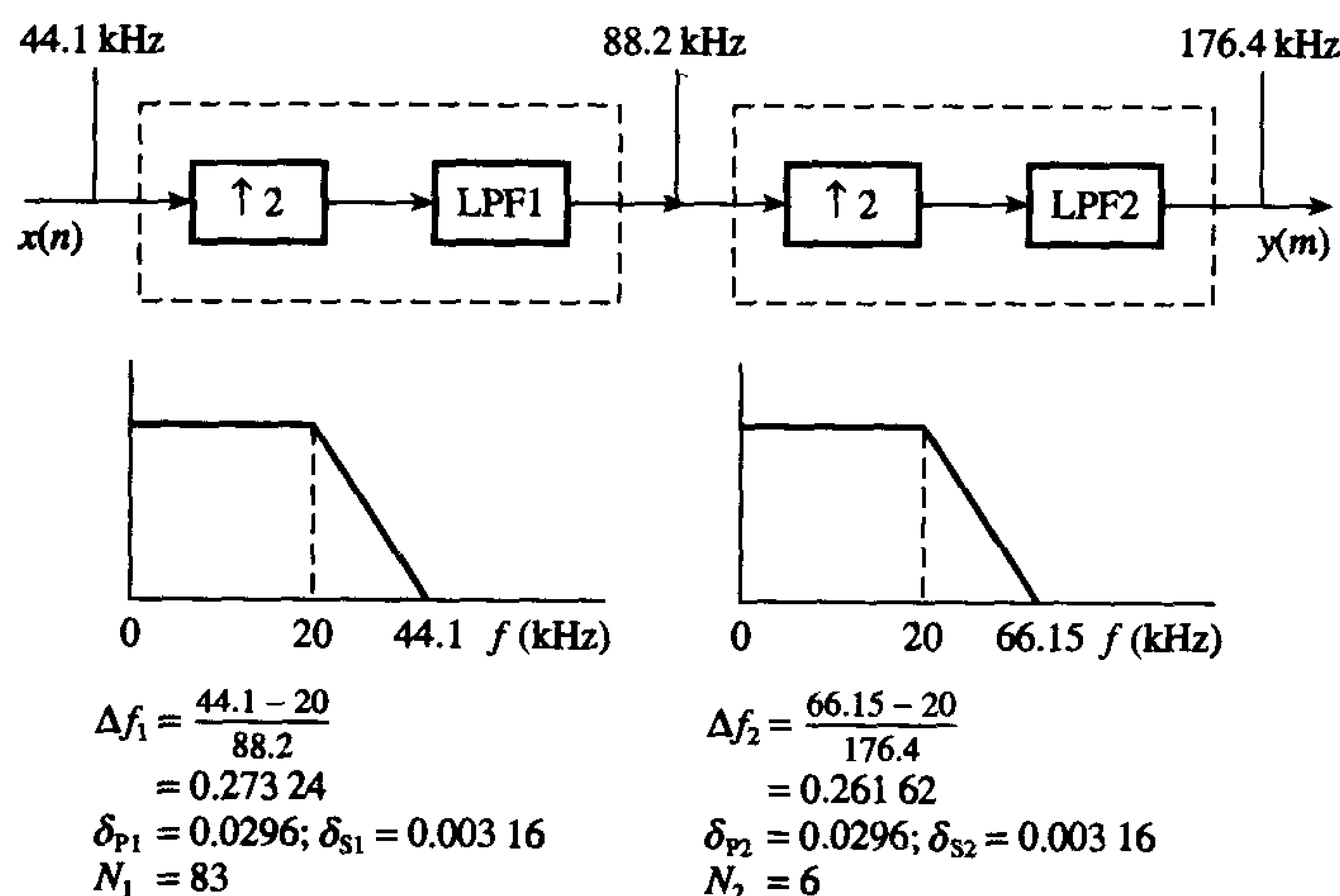


图9.27 满足例9.5的一个二级内插滤波器

9.7.3 高质量数据获取中的应用

在几乎所有的真实数据获取中，为了使频谱混叠较小，通常采用一种相对复杂的抗混叠滤波器。在一个多通道系统中，每个模拟通道必须配有一个单独的抗混叠滤波器，且它们不能互相复用。在需要很多模拟通道的场合（例如，生物医学可能需要32个通道），使用模拟抗混叠滤波器变得非常昂贵。通过使用数字抗混叠滤波器，每个通道的模拟滤波器可以被一个很简单的滤波器代替，从而在很大程度上降低了成本。另外，可以避免对模拟滤波器的严格相位要求，且模拟抗混叠滤波器支持多抽样频率的困难（每个抽样频率需要一个不同的截止频率）也可以克服。

图9.28给出了一个多抽样率数据获取系统（Quarmby，1984）的框图。通过前端的RC滤波器过抽样输入信号，再利用多抽样率技术降低抽样率到希望的抽样频率，从而达到所要求的混叠电平。需要付出的主要代价是ADC必须工作在一个较高的速率。

为了强化本节的内容，使大家更好地理解使用数字化抗混叠滤波器的益处，我们通过一个例题来讨论其实际应用。

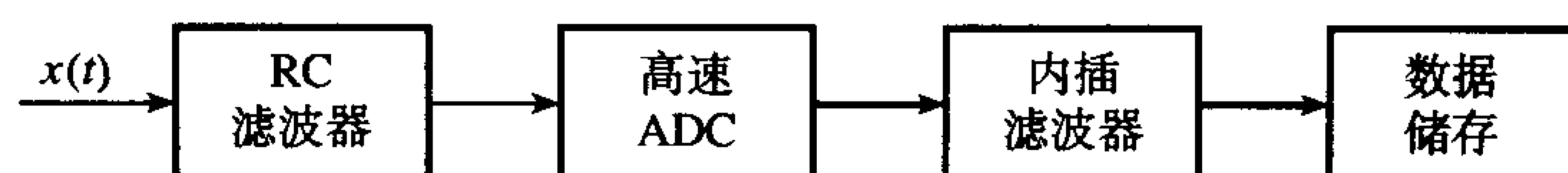


图9.28　一个简单的多抽样率数据获取系统

例9.6　一个普通的需求是，为采集生理学数据多通道（最大有32个）的数据获取系统。每个模拟通道是用户独自标定的，截止频率从0.5 Hz到200 Hz，并可选择从1 Hz到1000 Hz的抽样频率。每个通道总的滤波器指标是

通带波纹	$\leqslant 0.5$ dB
信号－混叠比	$\geqslant 45$ dB（在通带）
通带边沿频率	$0.5\ \text{Hz} \leqslant f_p \leqslant 200\ \text{Hz}$
阻带边沿频率	$\leqslant 3f_p$

幅度和相位失真都要尽可能低。为了减少元件个数，降低PCB的尺寸和费用，前端只能使用简单的模拟滤波器。

解：

单独应用模拟的抗混叠滤波器要求一个很高阶数的滤波器。一个替换方法是在每个通道使用一个简单的、同一的滤波器，在一个共同的固定抽样率上进行过抽样，再抽取到希望的抽样率上。在每一级我们必须保证指标得到满足。

可以在每个通道使用一个简单的单极点RC滤波器，但这会要求一个非常高的抽样率。我们将使用一个二阶巴特沃斯滤波器，因为它能满足我们在生物医学工程中的要求。

一个二阶巴特沃斯滤波器的幅频响应如下：

$$A(f) = \frac{1}{[1 + (f/f_c)^4]^{1/2}}$$

在图9.29(a)中给出其图形。从图中可以看出在0到f_c频带内存在一个明显的误差。为了使误差在滤波器的允许范围之内，感兴趣的最高频率（这个例子是200 Hz）应该刚好在f_c以下。这样得到f_c的值为

$$20\log[1 + (200/f_c)^4]^{1/2} \leqslant 0.5\ \text{dB}$$

解得$f_c \geqslant 338.39$ Hz。为了方便和容许后级引入的附加误差，使用了$f_c = 500$ Hz。500 Hz的f_c得到一个在200 Hz处下降到0.11 dB的幅频响应。

下面，我们建立一个所有通道共用的抽样频率。图9.29(b)给出了用巴特沃斯滤波器带限每个通道和抽样之后的信号频谱（假定信号是宽带的）。根据图示，可以看出我们需要一个抽样频率F_s，使得在f_p（$f_p = 200$ Hz）处混叠电平至少低于信号电平45 dB：

$$20\log\{1 + [(F_s - 200)/500]^4\}^{1/2} \geqslant 45\ \text{dB}$$

解得$F_s \approx 6.67$ kHz。在抽取期间，为了方便起见，令$F_s = 8192$ Hz。因而，一个适合一般需要的抽取滤波器的总特性如下：

输入抽样频率	8.192 kHz
输出抽样频率	1 Hz < F_s < 1000 Hz
阻带衰减	50 dB
通带波纹	0.01 dB
通带边沿频率	0.5 Hz < f_p < 200 Hz
抽取因子	8.192 < M < 8192
阻带边沿频率	< $2f_p$

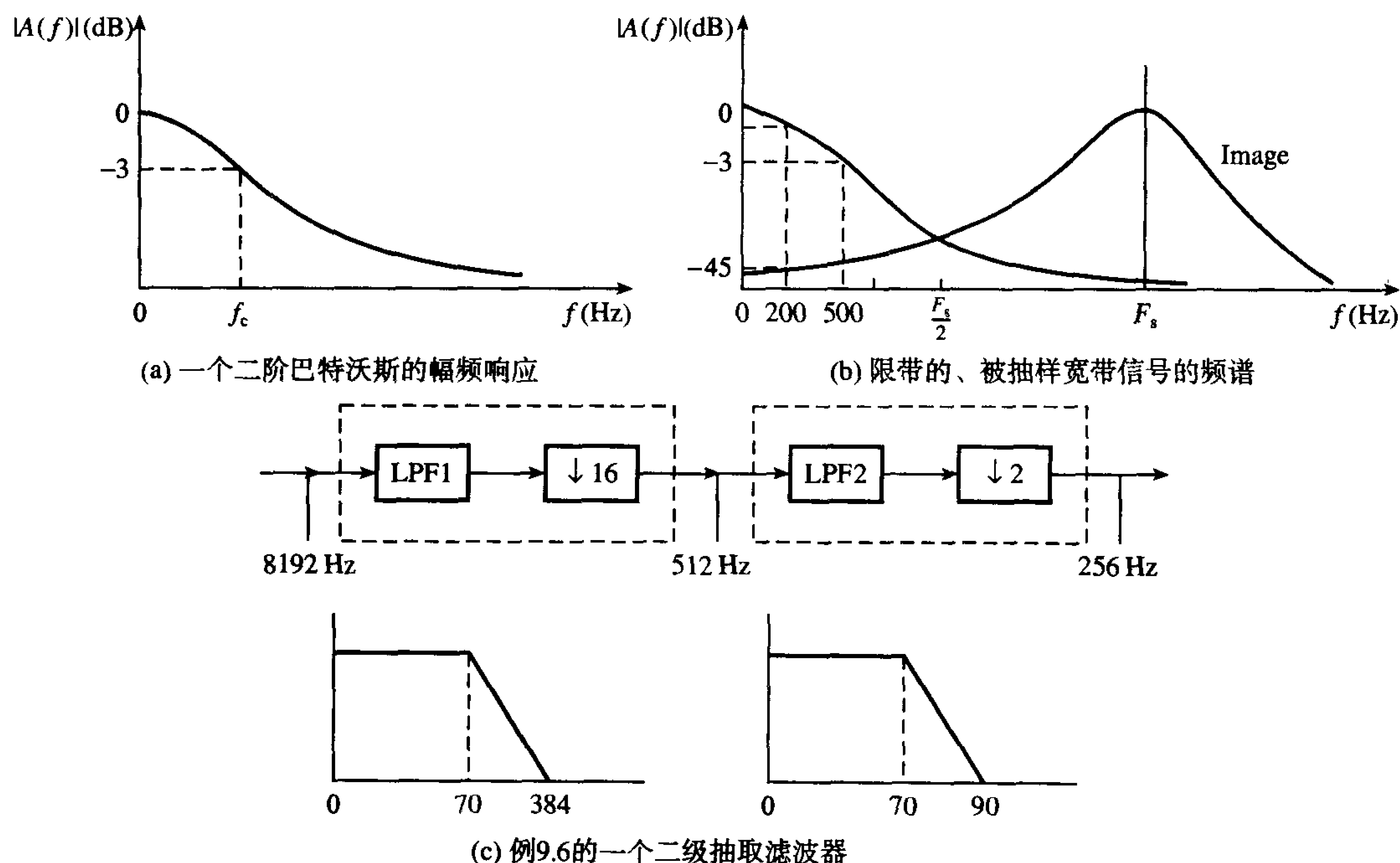


图 9.29　例 9.6 的参考图

为简便起见，我们给抽样频率 F_s 设定一个用户可选的限制，即只用整数因子来抽取。如果处理器容量足够我们进行非整数因子抽取，则这个限制可以取消。这样，可能的抽样频率和它们对应的抽取因子如下：

M	8	16	32	64	128	256	512	1024	2048	4096	8192
F_s (Hz)	1024	512	256	128	64	32	16	8	4	2	1

从概念上讲，我们可以设想一个包括了 11 级抽取滤波器的系统，对某种特别指标只需选择其中某级输出即可。

作为示例，让我们考虑一个收集脑电图（EEG）信号的系统。用户对各个通道的要求是

抽样频率	256 Hz
阻带衰减	45 dB
通带波纹	0.5 dB
通带	0 ~ 70 Hz

以上技术指标转换成抽样率转换器的技术指标，与以上通用抽取滤波器的指标一致：

输入抽样频率	8192 kHz

输出抽样频率	256 Hz
抽取因子	32
阻带衰减	50 dB
通带波纹	0.01 dB
通带	0 ~ 70 Hz
阻带	90 ~ 128 Hz

使用设计程序（指导手册的 CD 中），一个有效的抽取滤波器（在考虑计算和系统复杂性基础上）是如图 9.29(c)所示的二级系统。

例 9.7

(a) 一个数字处理系统，其输入是从 0 ~ 20 kHz 的模拟信号，使用过抽样技术和一个一阶 Σ-Δ 调制器来把模拟信号变换成抽样率为 3.072 MHz 的数字信号流。Σ-Δ 调制器的 z 复平面模型如图 9.30 所示。确定由于过抽样和噪声频谱整形而在信号量化噪声比（SQNR）上的总增益，进而估计转换器的有效位数或比特。

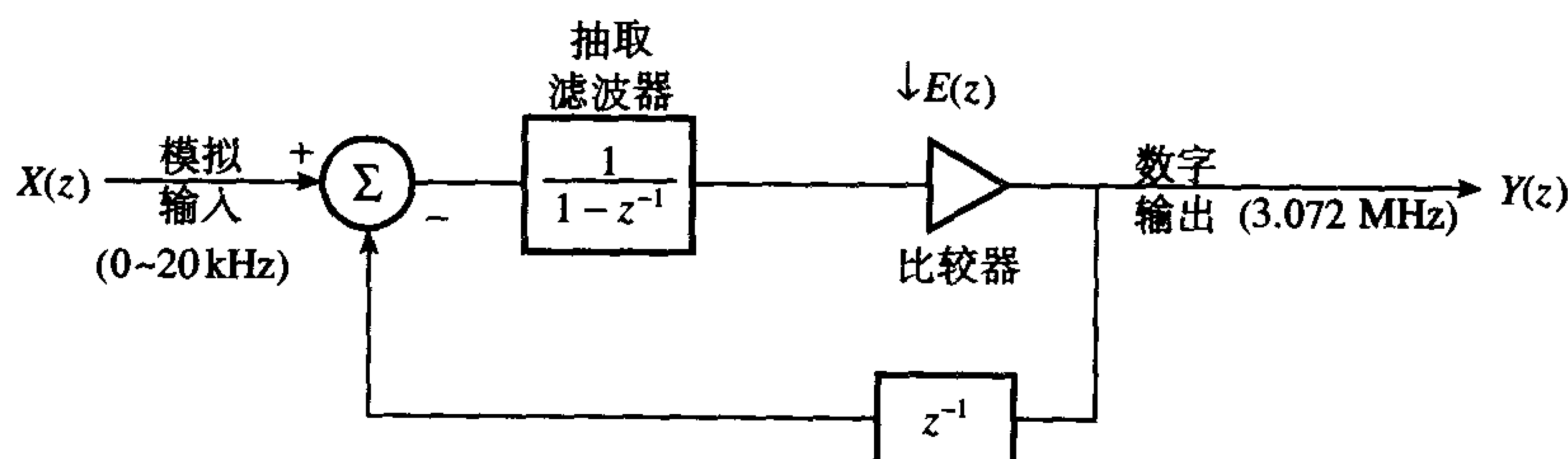


图 9.30　δ-Δ 调制器的 z 复平面模型

(b) 在框图层次上设计一个二级抽取滤波器，将图 9.30 中 Σ-Δ 调制器的输出从一个 3.072 MHz 的单比特流变换成48 kHz的多比特流。抽取滤波器的通带和阻带波纹分别为0.001和0.0001。

你的设计必须包含以下内容：

- 确定总的抽取和内插因子，且附带一个可调的指示；
- 确定二级抽样率变换器的内插和抽取因子对，且仔细分析它们的计算和存储复杂度来支持你的选择；
- 确定抗混叠和取镜像滤波器的边沿频率、长度、通带和阻带波纹。

注意，你可以假定滤波器是直接形式的 FIR，且其长度由下式给出：

$$\text{滤波器长度}, N = \frac{-10\log(\delta_p\delta_s) - 13}{14.6\Delta f} + 1$$

其中

Δf = 归一化过渡带宽

解：

(a) 噪声传递方程为

$$N(z) = 1 - z^{-1}$$

幅频响应为

$$\begin{aligned}|N(z)|_{z=e^{j\omega T}} &= |(1-e^{-j\omega T})| \\ &= [(1-\cos\omega T)^2+\sin^2\omega T]^{\frac{1}{2}}\end{aligned}$$

在$f = 20$ kHz，$F_s = 3.072$ MHz处$\omega T = 2.3438°$和$|N(e^{j\omega T})| = 0.0409$，这相当于量化噪声被消减了27.76 dB。

ADC的有限字长主要由过抽样和噪声整形的组合SQNR确定，这里等于18.85 + 27.76 = 46.61 dB，对应着大约7个比特的有效ADC分辨率（即SQNR = 6.02 dB + 1.77 dB）。

(b) 根据输入和输出抽样率的比值，总的抽取因子是64。对这个抽取因子，二级结构共有3种可能的因子组合：8 × 8、16 × 4、32 × 2。计算复杂度考虑存储需求及每秒乘法次数MPS。

对8 × 8抽取滤波器，二级子抽取滤波器的输出抽样率分别为384 kHz和48 kHz。第一级抗混叠滤波器的带边沿频率是0、20 kHz和360 kHz，其归一化过渡带宽为0.1106。通带和阻带波纹分别为0.001/2 = 0.0005和0.0001。根据以上滤波参数得到$N_1 = 38$。对第二级，带边沿频率是0、20 kHz和24 kHz，其归一化过渡带宽为0.0104，滤波器长度$N_2 = 396$。

同样地，对16 × 4结构，输出抽样率分别为192 kHz和48 kHz。第一级抗混叠滤波器的带边沿频率为0、20 kHz和168 kHz，过渡带宽为0.048 17，$N_2 = 198$。对32 × 2结构，输出抽样率分别为96 kHz和48 kHz。第一级抗混叠滤波器的带沿频率为0、20 kHz和72 kHz。过渡带宽为0.0169，$N_1 = 244$；第二级滤波器带沿频率为0、20 kHz和24 kHz，过渡带宽为0.041 66，$N_2 = 100$。

各个结构的计算复杂度如下：

$M_1 \times M_2$	存储量	MPS
8 × 8	434	33.6×10^6
16 × 4	284	26.01×10^6
32 × 2	384	28.82×10^6

根据比较，16 × 4结构是最有效的，其框图参见图9.31。

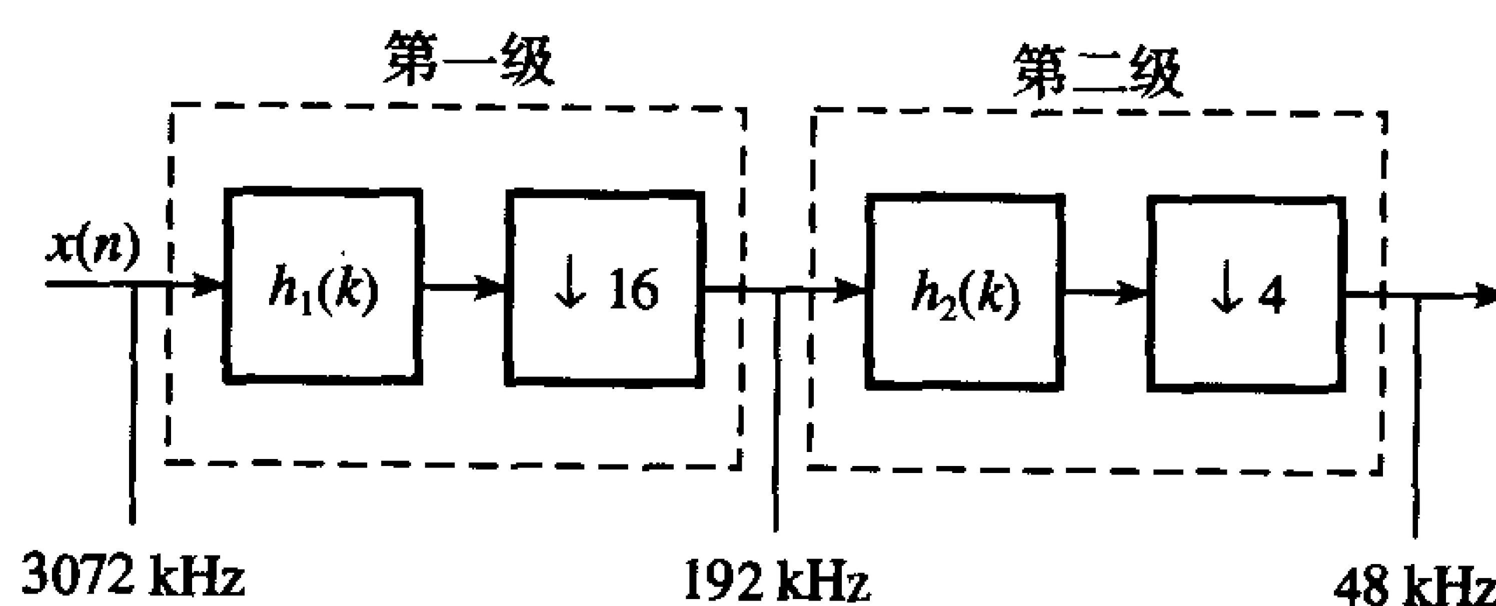


图9.31 例9.7(b)的16 × 4结构的框图

9.7.4 多抽样率窄带数字滤波

窄带数字滤波器的特点是通带与阻带间尖锐的过渡带，且与抽样频率相比，通带是一个很小的量。因此窄带FIR滤波器通常需要很多的系数，给设计和实现这种滤波器带来很大的困难，因为它们对有限字长效应非常敏感（例如舍入噪声和系数量化噪声）。另外，还需要大量的存储单元和计算能力。多抽样率方法能克服这些困难，使FIR滤波器的计算性能达到椭圆IIR滤波器的水平。

图9.32给出了一个简单的多抽样率滤波结构。输入序列的抽样频率首先被抽取滤波器尽可能地降低。使所希望的滤波器操作在一个较低的抽样频率上进行。最后，滤波后数据的抽样频率被内

插滤波器还原到最初的状态。在抽取滤波器和内插滤波器上使用相同的抽样率变换因子，保证了输入信号 $x(n)$ 与输出信号 $y(n)$ 具有相同的抽样率。

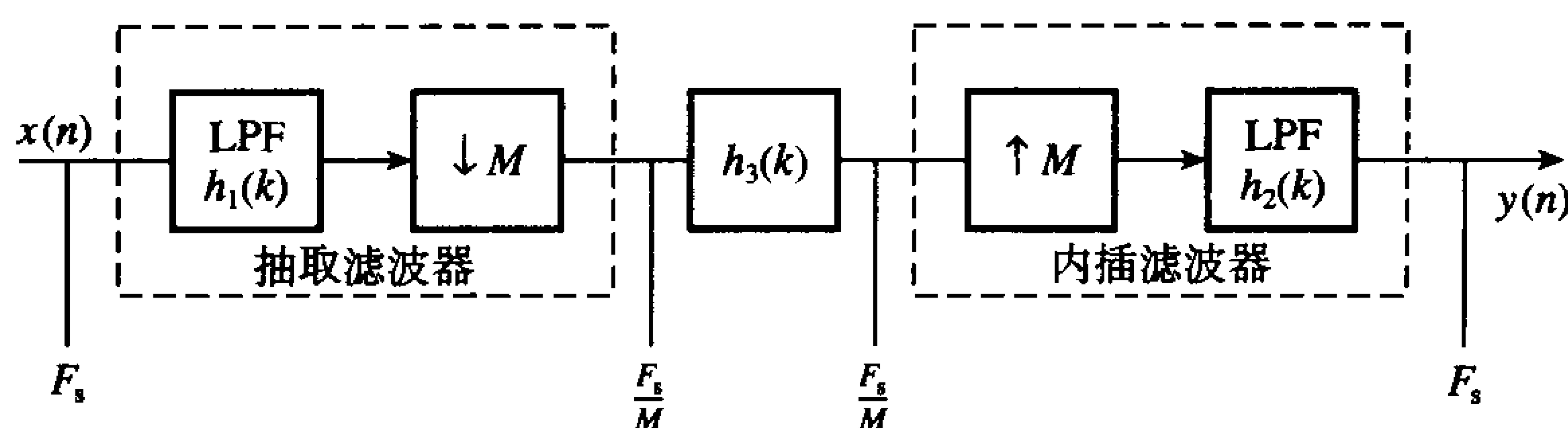


图 9.32　多抽样率窄带滤波

9.7.4.1　窄带低通和带通滤波

对于窄带低通滤波，图 9.32 的滤波器 $h_1(k)$ 和 $h_2(k)$ 可以均为低通滤波器，而 $h_3(k)$ 是不必要的。设计目标在于使图 9.32 结构的总输入－输出特性与所希望的常规低通滤波器等效。当然，由于频谱混叠和镜像，实际结果并不与常规低通滤波器完全一致。在实践中，为保证滤波器总特性满足需要，滤波器 $h_1(k)$ 和 $h_2(k)$ 是完全相同的，都分别具有通带波纹 $\delta_p/2$ 和阻带波纹 δ_s，δ_s 和 δ_p 与低通滤波器的通带和阻带失真等效。

多抽样率带通滤波设计需要考虑更多的东西。除非你想设计某种称为整数分之一的带通滤波器，即系统的带边沿频率恰好是最低抽样频率的倍数 $F_s/2M$。在这些情况下，抽取 / 内插因子 M 及滤波器的带边沿频率满足下面的条件（Crochiere and Rabiner, 1983）：

$$M = F_s/2(f_{su} - f_{sl}) \tag{9.17a}$$

$$f_{sl} = kF_s/M,\ k\text{是一个整数},\ 0 < k < M-1 \tag{9.17b}$$

$$f_{su} = (k+1)F_s/M \tag{9.17c}$$

其中 f_{sl} 和 f_{su} 分别是下、上阻带的边沿频率。9.17a 式给出可能的最大抽取因子，9.17b 式和 9.17c 式确定了下、上阻带的边沿频率和带数 k。

一个简单但不很有效的替代方法是，多抽样率带通滤波先使用适合的低通滤波器将数据尽可能地抽取，再带通滤波低抽样率信号，最后内插还原到希望的抽样率，如图 9.33 所示。很显然，必须防止希望的通带被抽取 / 内插过程的混叠和镜像效应所干扰。

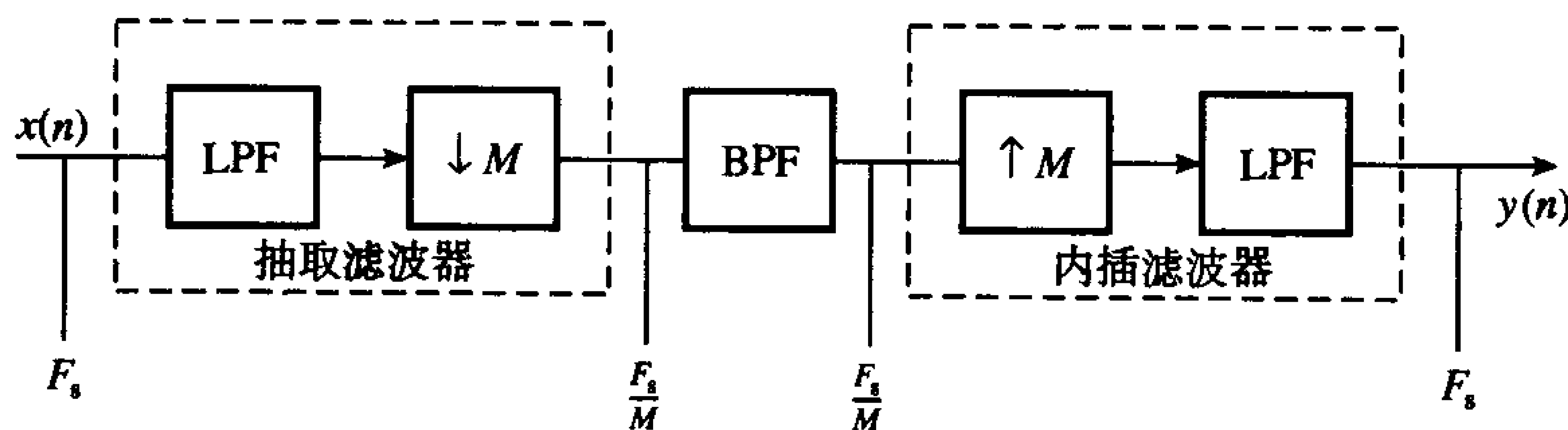


图 9.33　多抽样率窄带滤波的一种实现方法

9.7.4.2　窄带高通和带阻滤波器

窄带高通和带阻滤波器可以分别通过低通和带通滤波器的互补结构来实现：

$$H_{hp}(w) = 1 - H_{lp}(w) \tag{9.18a}$$

$$H_{bs}(w) = 1 - H_{bp}(w) \tag{9.18b}$$

高通和带阻滤波器结构如图 9.34 所示。例如对高通滤波器，信号 $x(n)$ 先被低通滤波。再从原始未滤波信号中减去滤波后数据。减法操作前信号 $x(n)$ 必须先经过一个延迟，延迟量等于低通滤波器产生

的输出延迟。显然，应该尽可能使信号经过低通滤波器的延迟是抽样时间的整数倍。低通滤波器必须具有正确的通带和阻带要求才能得到希望的高通滤波器。

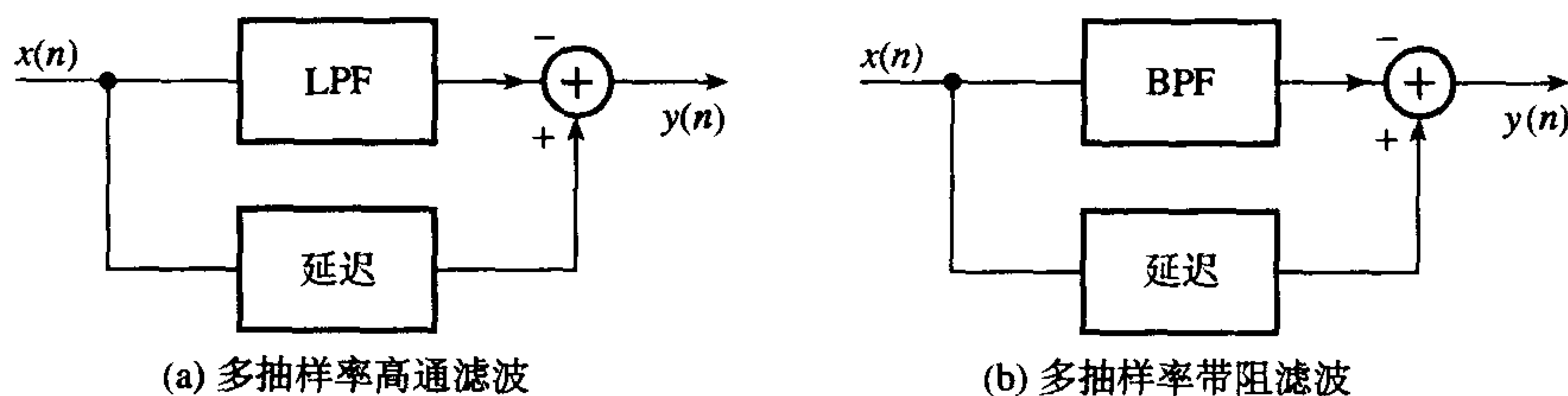


图 9.34 使用低通/带通的互补来实现多抽样率高通/带阻滤波

例9.8 在与胎儿护理有关的研究中，需要评估胎儿心电测量系统对胎儿心率的电气活动的影响（ECG 或心电图）（Westgate et al., 1990）。为此要求 ECG 的某些特征被量化。包括信号的电源频率干扰（50 Hz），因为在电源频率处附近存在信号能量，因此需要一个很窄的带通滤波器。滤波器指标为

通带	49 ~ 51 Hz
阻带边沿频率	47 Hz 和 53 Hz
阻带衰减	30 dB (δ_p = 0.031 62)
通带波纹	0.1 dB (δ_s = 0.011 579)
抽样频率	500 Hz

解：

如果直接设计上面的滤波器，9.3 式给出了一个长达 4018 个系数的 FIR 滤波器，这不能应用于实际情况中。

使用多抽样率方法时，存在许多选择方案。一种是先将数据抽取到尽可能低的抽样率（符合上面滤波器的要求，可参见例 9.6）。在这种情况下，最低抽样率为 125 Hz，使我们仍能保留 0 ~ 62.5 Hz 的频带。总的抽取滤波器指标为

通带波纹	0.05 dB (δ_p = 0.005 789 5)
阻带衰减	30 dB (δ_s = 0.031 62)
通带	0 ~ 53 Hz
输入抽样频率	500 Hz
输出抽样频率	125 Hz

抽取因子是 4。使用指导手册的 CD 上多抽样率设计程序（参见前言），设计二级抽取滤波器（参见图 9.35）。最优法（参第 7 章）用于获得图 9.35 两个滤波器的系数。ECG 数据经过滤波器和 9.4 节描述的多级抽取程序抽取。抽取前后的示例数据如图 9.36 所示。

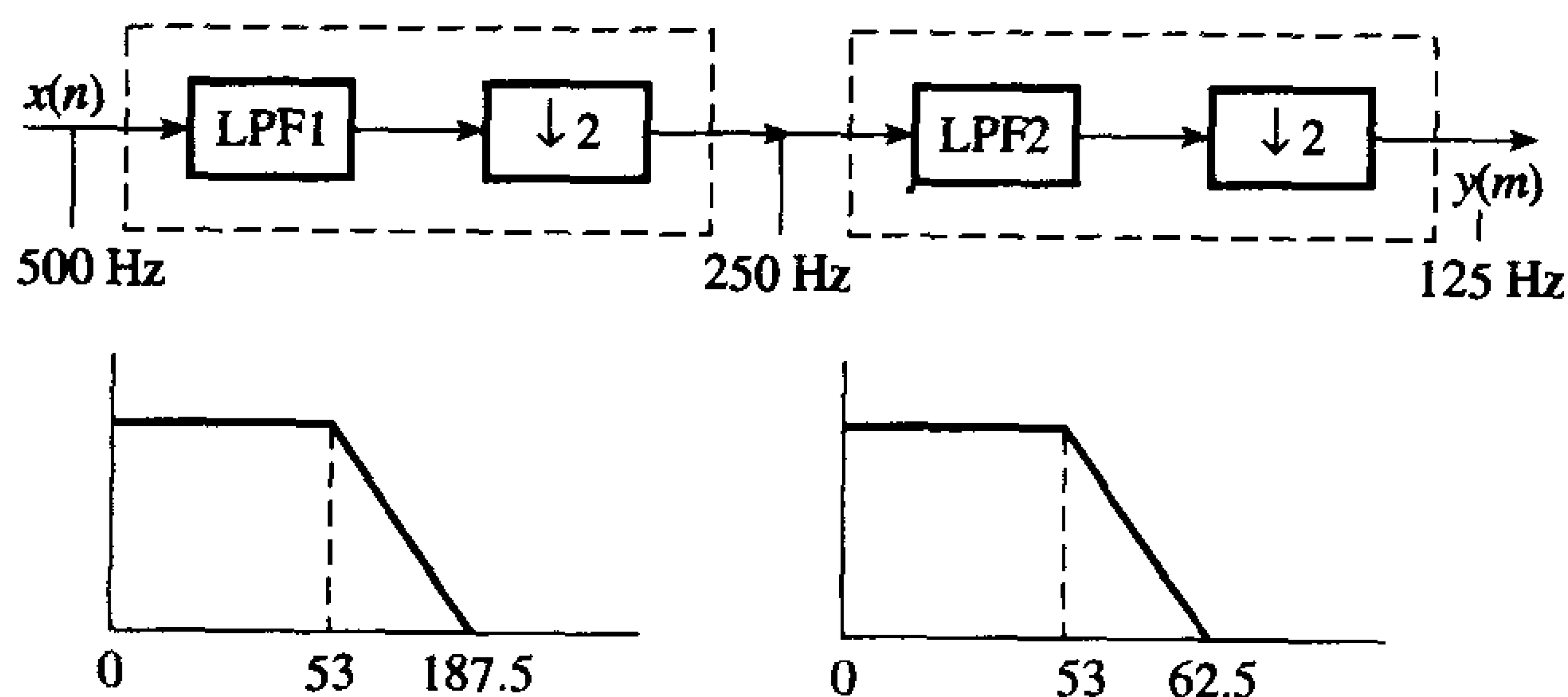
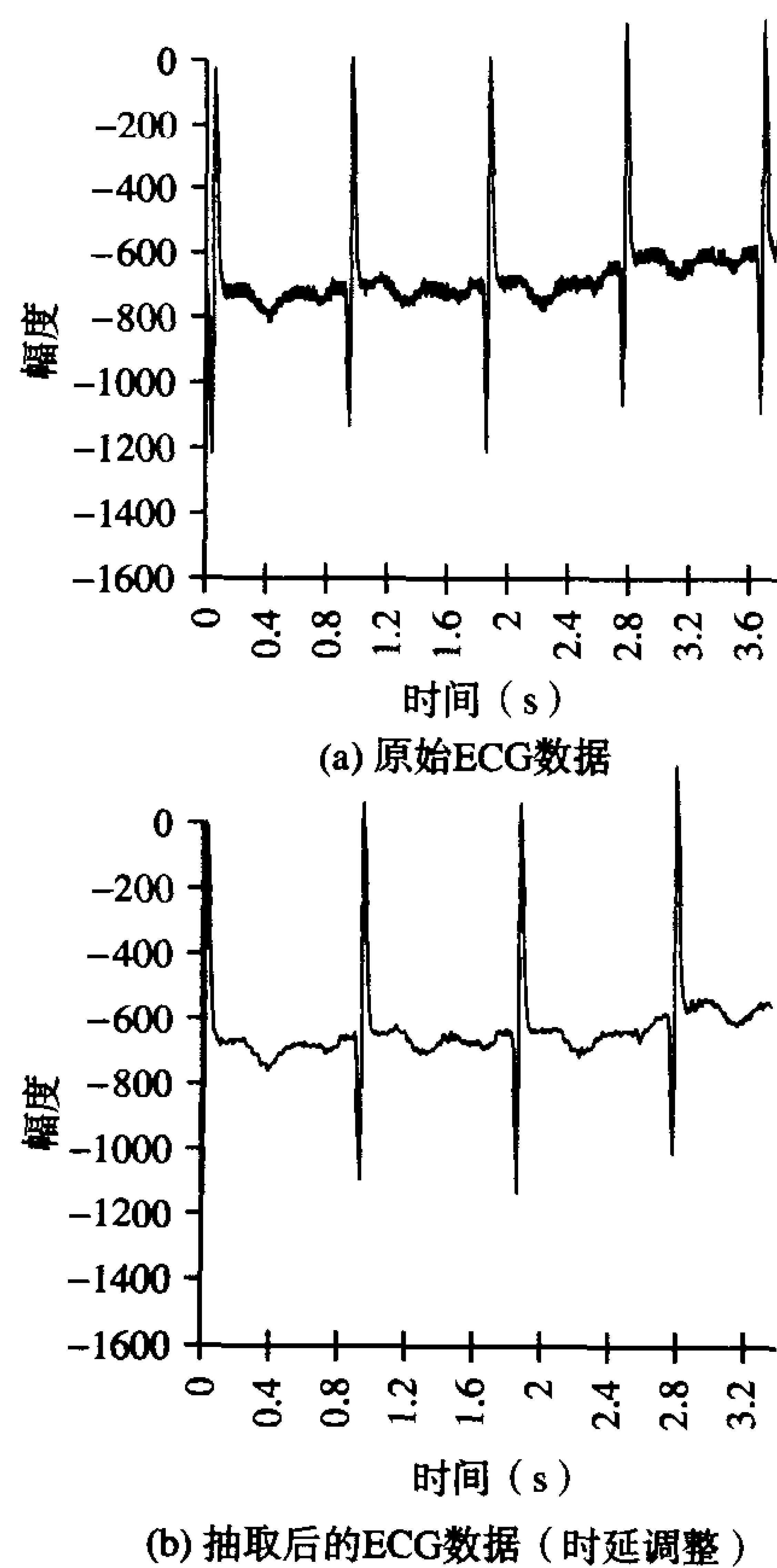


图 9.35 降低 ECG 数据抽样率的抽取滤波器



(a) 原始ECG数据

(b) 抽取后的ECG数据（时延调整）

图 9.36　抽取前后的 ECG 数据

一个适合例题要求（但工作于新的、抽样率降低的数据）的滤波器被设计出来，在本例中，对于主滤波器（$\delta_p = 0.005\ 789\ 5$，$\delta_s = 0.031\ 62$），滤波器系数的个数为 113.4。滤波后，可以将数据重新内插还原到它的初始抽样率。

例 9.9　为提取胎儿 ECG 中的基线偏移（baseline shift）设计一个合适的多抽样率低通滤波器。滤波器应满足以下指标：

通带	0 ~ 0.4 Hz
阻带	0.5 ~ 250 Hz
通带波纹	0.01
阻带波纹	0.001
抽样频率	500 Hz

解：

我们采用的方法是先将抽样率降低到 1 Hz，再内插还原到 500 Hz。在本例中，抽取滤波器的总指标为

通带波纹	0.01
阻带波纹	0.001
通带	0 ~ 0.4 Hz
阻带边沿频率	0.5 Hz

抽样频率	500 Hz
抽取因子	500

使用指导手册CD上的多抽样率设计程序，可以得到直到四级的各种不同的抽取滤波器，我们感兴趣的特性在表9.6中列出。考虑到实现的复杂度，图9.37给出的三级抽取滤波器被选为最佳方案。再使用最优设计程序（第7章），计算出抽取滤波器的滤波器系数。使用多级抽取，ECG数据的抽样率被降低到1 Hz，再内插还原成500 Hz。

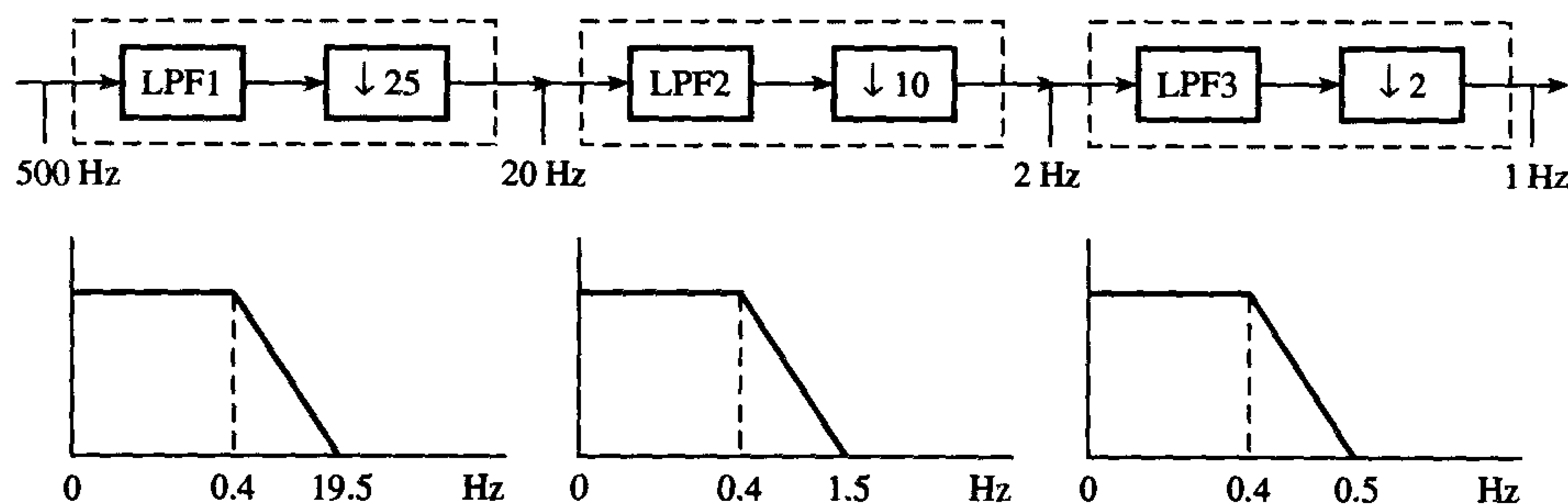


图9.37　例9.9的三级抽取滤波器

表9.6　有效的抽取滤波器

级数	MPS	TSR	M_i	滤波器长度 N_i	通带边沿频率 f_p(Hz)	阻带边沿频率 f_s(Hz)	归一化过渡带宽 Δf_i	通带波纹	阻带波纹
1			500	12 707	0.4	0.5	0.002 2	0.01	0.001
2	1807	430	50	153	0.4	9.5	0.018 20	0.005	0.001
			10	277	0.4	0.5	0.01	0.005	0.001
3	1705	189	25	77	0.4	19.5	0.038 20	0.0033	0.001
			10	53	0.4	1.5	0.055	0.0033	0.001
			2	59	0.4	0.5	0.05	0.0033	0.001
4	1444	172	2	2	0.4	249.5	0.498 2	0.0025	0.001
			25	83	0.4	9.5	0.036 40	0.0025	0.001
			5	27	0.4	1.5	0.110 00	0.0025	0.001
			2	60	0.4	0.5	0.050 0	0.0025	0.001
4	1724	169	25	79	0.4	19.5	0.038 20	0.0025	0.001
			2	3	0.4	9.5	0.455 00	0.0025	0.001
			5	27	0.4	1.5	0.110 0	0.0025	0.001
			2	60	0.4	0.5	0.050 0	0.0025	0.001

9.7.5　高分辨率窄带频谱分析

在11章我们将论述FFT的一项重要应用，即信号的频谱估计。FFT能够给出信号0到二分之一抽样频率平均间隔的频谱分量。在许多应用如声呐、地震学、雷达、生物医学和振动分析中，所需要的信号可能只占获取数据频谱的一个很窄的部分。这时，直接使用FFT会导致很大且不必要的计算量。多抽样率技术可以在应用FFT前把感兴趣的频带分离和转换到低频段上，使计算量大大减轻。或者我们也可以在分辨率和计算量之间做一种权衡。

在降低抽样率的数据上进行FFT运算，可以在减轻计算量基础上得到同样的分辨率；或者在同样计算量（原始序列直接FFT）的情况下得到更高的分辨率。降抽样率技术允许我们能在一个更大的尺度上观察窄带频谱。

窄带频谱分析使用多抽样率技术实质上是我们先前讨论窄带滤波的一种扩展，也有着同样的限制条件。信号首先被带通滤波以分离出感兴趣的频带，然后将滤波后信号的抽样频率通过抽取降低

到F_s/M，这里F_s是信号$x(n)$的抽样频率。最后应用FFT计算已经降低抽样率的序列$y(n)$的频谱。一个校正因子用于补偿由于$h(n)$的通带波纹造成的频谱误差。如果感兴趣的频带不满足条件，则可以使用一个包含所需频带的稍宽频带来解决。另一种是使用Liu和Mintzer（1978）提出的方法，即采用计算机搜索允许的抽取因子。

9.8 小结

处理多于一种抽样率的数字系统称为多抽样率系统。多抽样率系统的两个关键器件是抽取滤波器和内插滤波器。抽取滤波器允许我们以整数因子M或有理数因子L/M（$L<M$）有效地降低信号的抽样率。内插滤波器允许我们以整数抽取因子L或有理因子L/M（$L>M$）有效地增加信号的抽样率。

在实践中，考虑到最大的计算和存储效率，抽样率变换通过二级或更多级结构来实现。在多级设计中，单级滤波器具有宽松的要求，带来很少的滤波器系数，因而也就降低了对有限字长效应的敏感性。设计抽样率变换器的一种实用方法在本章中得到了详细介绍。

多抽样率系统的主要优势在于它们能够充分挖掘DSP的潜力，特别是使用DSP来带限一个接近奈奎斯特频率的信号而不用违反抽样定理的限制，且具有相当的衰减量。这些优势使它广泛应用于如CD播放器、数字滤波、数据获取和高分辨率数据获取系统。这些系统及其应用多抽样率器件的设计大多在文中已经过详细介绍。

本书指导手册的CD上提供了一套C语言程序（参见前言），能够设计和实现多抽样率系统。使用MATLAB的多抽样率DSP在附录9B中给出。

习题

9.1 一个一级抽取滤波器的特性如下：

抽取因子　　3

抗混叠滤波器参数

$$h(0)=-0.06=h(4)$$
$$h(1)=\ 0.30=h(3)$$
$$h(2)=\ 0.62$$

输入$x(n)$数据的连续值为{6, −2, −3, 8, 6, 4, −2}，计算并列出滤波后输出$w(n)$及抽取滤波器的输出$y(m)$。

9.2 (a) 一个用于将抽样率从96 kHz降低到1 kHz的三级抽取滤波器的框图如图9.38所示。假定抽取因子依次为8、4和2，指出在每级输出的抽样率。

(b) 假定(a)的抽取滤波器满足以下指标：

输入抽样频率F_s	96 kHz
抽取因子M	96
通带波纹	0.01 dB
阻带衰减	60 dB
感兴趣频带	0~450 Hz

确定每级抽取滤波器的带边沿频率。

(c) 假定一个抽取滤波器的输入和输出抽样率分别为96 kHz和1 kHz：

(i) 写出总的抽取因子；

(ii) 写出二级结构抽取因子的所有可能的整因数集（降序排列）；

(iii) 按三级结构重做(ii)；

(iv) 按四级结构重做(ii)。

(d) 对(a)中的抽取滤波器，计算每秒的总乘法次数（MPS）和总存储需求（TSR）。

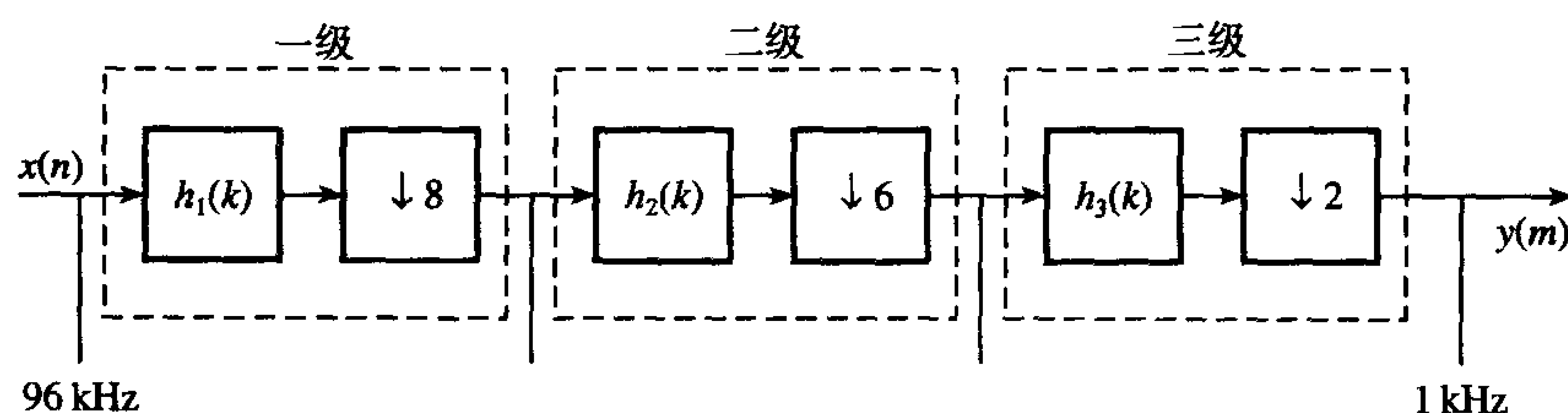


图 9.38 习题 9.2 的抽取滤波器框图

9.3 (a) 在框图层次上设计一个二级抽取滤波器，将一个音频信号按因子 32 降低抽样率且满足以下指标。你的回答必须指出一对合适的抽取因子。分析它的计算及存储复杂度以支持你的选择。确定每级抽取的输入输出抽样率，以及设计的每级滤波器的下列参数：

带边沿频率

归一化过渡带宽

通带和阻带波纹

滤波器长度

可以假定滤波器是直接 FIR 形式，其长度由下式确定：

滤波器长度，$N = \dfrac{-10\log(\delta_p\delta_s) - 13}{14.6\Delta f} + 1$

其中

Δf = 归一化过渡带宽

(b) 采用适当的略图说明感兴趣频带（0 ~ 3.4 kHz）被抽取滤波器保护而免于混叠。

输入抽样频率 F_s	256 kHz
数据中感兴趣的最高频率	3.4 kHz
通带波纹 δ_p	0.05
阻带波纹 δ_s	0.01

9.4 为一个高质量数据获取系统设计一个抽取滤波器，其总指标如下：

音频	0 ~ 20 kHz
输入抽样频率	3.072 MHz
输出抽样频率	48 kHz
通带波纹	< 0.001 dB
阻带衰减	> 86 dB

9.5 需要计算存在于宽带信号中的一个窄带信号频谱。感兴趣频带为 49 ~ 51 Hz，但混合信号包含了 0 ~ 100 Hz 的频带。现在获得了按 1 kHz 抽样的混合信号的 N 点数据序列 $x(n)$。

(1) 图示出怎样使用多抽样率方法得到希望的信号频谱。

(2) 评估多抽样率方法比直接 FFT 方法的计算优势。比较两种方法得到的频谱分辨率。

9.6 需要一个高质量、有效的窄带滤波器以分离和估计一个信号的主成分。滤波器应满足以下指标：

通带	49 ~ 51Hz
阻带边沿频率	48 Hz 和 52 Hz
阻带衰减	60 dB
通带波纹	0.01 dB
抽样频率	1000 Hz

使用多抽样率方法设计适合的滤波器。

9.7 需要解释一个特定的生理信号，其抽样率为 256 Hz。为此需要分离和分析每个频带的时域/频域特征。第一步是设计一个合适的多抽样率系统来把信号分割成以下频带：

0.5 ~ 4 Hz

4 ~ 8 Hz

8 ~ 13 Hz

13 ~ 16 Hz

多抽样率系统不应在每个频带内产生大于0.01 dB的波纹，而对频带外应有至少50 dB的衰减。

9.8 一个 DSP 系统，输入是模拟的音频信号 0 ~ 20 kHz，使用过抽样技术和一个一阶 Σ-Δ 调制器将模拟信号变换成一个抽样率为3.072 MHz的数字比特流。Σ-Δ调制器的z平面模型如图 9.39 所示。

(a) 确定过抽样能带来的信号量化噪声比总增益和噪音频谱整形，由此估计变换器的有效分辨率（比特）。

(b) 在框图层次上设计一个二级抽取滤波器，将图9.39的Σ-Δ调制器输出从一个3.072 MHz的单比特流转换成48 kHz的多比特流。抽取滤波器的通带和阻带波纹分别是0.001和0.0001。

你的回答必须包括以下内容：

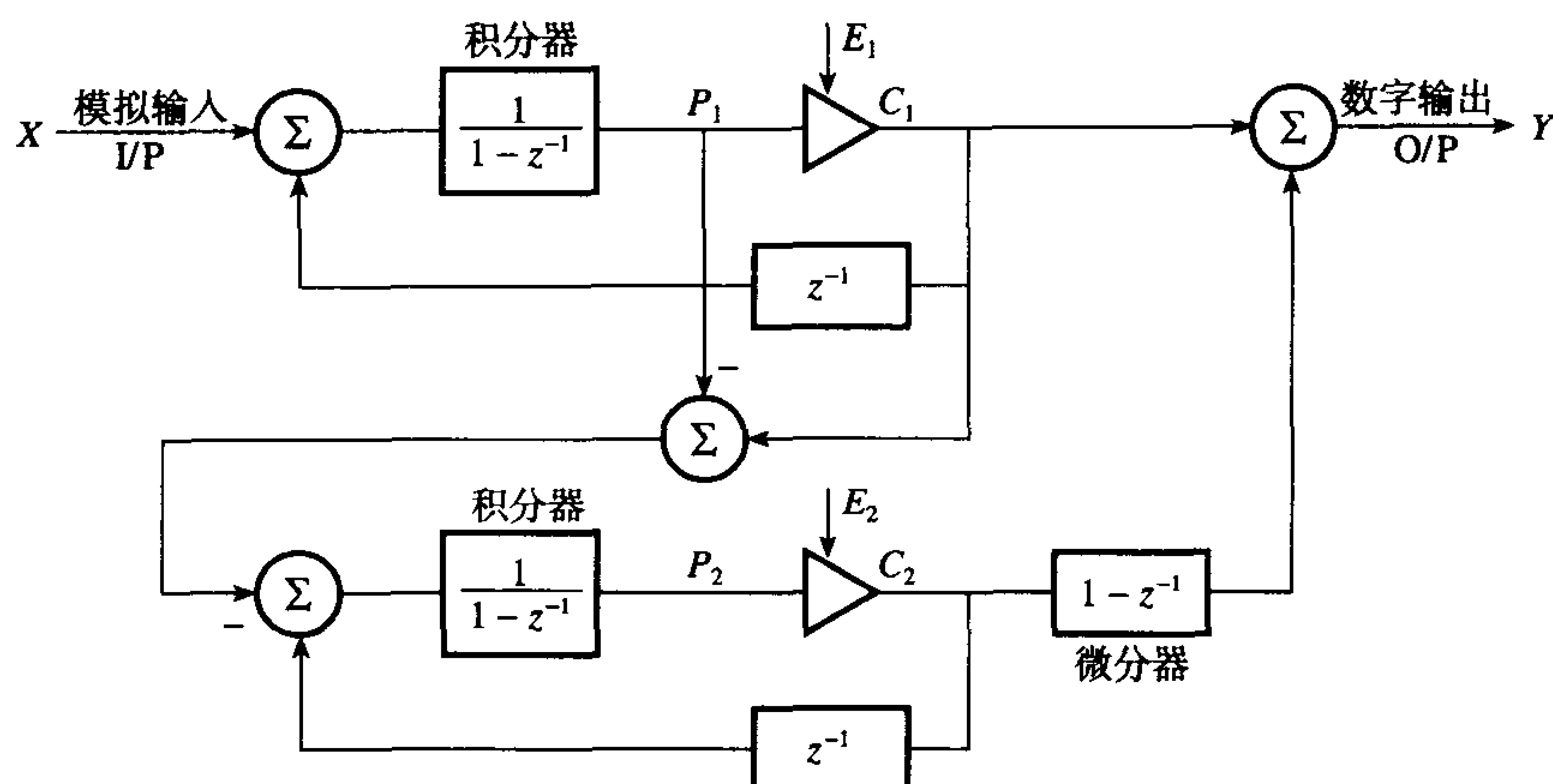
- 估值总的抽取和内插因子，附带如此确定的正当理由；
- 估值二级抽样率变换器的抽取和内插因子对，详细分析它们的计算和存储复杂度以支持你的选择；
- 确定抗混叠及去镜像滤波器的带边沿频率、长度、通带和阻带波纹。

可以假定滤波器是直接 FIR 形式，其长度由下式确定：

滤波器长度，$N = \dfrac{-10\log(\delta_p\delta_s) - 13}{14.6\Delta f} + 1$

其中

Δf = 归一化过渡带宽



注意：二阶Σ-Δ调制器的输出变换$Y(z)$为：$Y(z) = X(z)+E_2(z)(1-z^{-1})^2$，
变量具有通常的意义

图 9.39 习题 9.8 的参照图

9.9 (a) (i) 采用框图解释使用抽取滤波器/内插滤波器的多抽样率低通数字滤波。多抽样率数字滤波的主要缺点是什么？

(ii) 根据对一级多抽样率低通滤波器的分析，指出当抽取因子大于2，多抽样率方法的计算效率好于通常的固定抽样率滤波。

进一步扩展你的分析，指出随着低通滤波器的带宽减小，多抽样率滤波的计算效率增加。讨论这个特点的实际应用。

(b) 需要低通滤波器满足以下指标：

通带边沿频率	4 Hz
阻带边沿频率	6.25 Hz
通带波纹	0.001
阻带波纹	0.0001
抽样率	500 Hz

在框图层次上设计一个满足以上指标的有效的多抽样率滤波器，包括一个二级抽取滤波器和一个二级内插滤波器。你的回答应包含以下内容：

- 估值总的抽取和内插因子，并给出如此确定的正当理由；
- 估值二级抽样率变换器的抽取和内插因子对，详细分析它们的计算和存储复杂度以支持你的选择；
- 适当标明的多抽样率低通滤波器框图；
- 确定抗混叠及去镜像滤波器的带边沿频率、长度、通带和阻带波纹。

(c) 与普通的直接FIR实现相比，(b)中的多抽样率低通滤波器的计算复杂度。并探讨你的回答。

你可以假定滤波器是直接FIR形式，其长度由下式确定：

滤波器长度，$N = \dfrac{-10\log(\delta_p\delta_s) - 13}{14.6\Delta f} + 1$

其中

Δf = 归一化过渡带宽

9.10 (a) 解释在一个多抽样率处理系统中，下面二者角色的不同：

(i) 抽取的滤波器；

(ii) 抽取器

(b) 需要为音频多抽样率系统设计一个二级抽取滤波器，并满足以下指标：

输入抽样频率 F_s	96 kHz
抽取因子 M	96
数据中感兴趣的最高频率 f_p	450 Hz
通带波纹 δ_p	0.05
阻带波纹 δ_s	0.01

表 9.7　二级抽取滤波器的设计信息

#	M_1	M_2	N_1	N_2	MPS	TSR
0	2	48	2	2651	2 747 000	2653
1	3	32	6	1768	1 960 000	1774
2	4	24	10	1326	1 556 000	1336
3	6	16	17	885	1 157 000	902
4	8	12	24	664	952 000	688
5	12	8	38	443	747 000	481
6	16	6	53	333	651 000	386
7	24	4	88	222	574 000	310
8	32	3	131	167	560 000	298
9	48	2	254	112	620 000	366

注意

#	设计选择的标号
M_1, M_2	一级和二级的抽取因子对
N_1, N_2	一级和二级滤波器系数的个数
MPS	每秒乘法次数
TSR	总存储需求

根据工具软件对该设计问题的初步分析，产生了表 9.7 的信息。根据设计要求和表 9.7 提供的信息：

(i) 在框图层次上设计二级抽取滤波器。你的回答应包括合适的抽取因子对及支持的理由、在各级输入和输出的抽样频率；

(ii) 对各级抽取滤波器，确定滤波器的下列参数：

带沿频率

归一化过渡带宽

阻带和通带波纹

滤波器系数个数

(iii) 借助合适的框图，证明感兴趣频带（0 ~ 450 Hz）被抽取滤波器保护而免于混叠。

9.11 需要为多抽样率音频系统设计一个二级抽取滤波器，并满足以下指标：

输入抽样频率 F_s	96 kHz
抽取因子 M	96
感兴趣的最高频率 f_p	450 Hz

通带波纹 δ_p	0.05
阻带波纹 δ_s	0.01

根据工具软件对该设计问题的初步分析，产生了表9.7的信息。根据设计要求和表9.7提供的信息：

(a) 利用9.11式确定各级的最优抽取因子；

(b) 舍入抽取因子到最近的整数，得到正确的总抽取因子；

(c) 与习题9.10(b)比较结果。

MATLAB 习题

9.12 一个连续时间信号由下面的方程给出：

$$x(t) = A\cos(2\pi f_1 t) + B\cos(2\pi f_2 t)$$

其等效的离散时间信号 $x(nT)$ 是通过将连续信号按抽样率 $F_s = 1/T$ 得到的。

(a) 利用MATLAB的帮助，从连续信号产生1000个数据抽样。假定 $F_s = 5000$ Hz，$f_1 = 50$ Hz，$f_2 = 100$ Hz，$A = 2$ 和 $B = 1$。

(b) 利用MATLAB 函数 decimate 按因子 10 降低抽样率；

(c) 利用MATLAB 函数interp按因子4对抽取后数据再增加抽样率。将从步骤(a)~步骤(c)得到的离散时间信号按照合适的抽样个数，用stem 函数显示。评论你的结果。

9.13 利用函数 resample 重复习题9.12。将每种情况下 resample 函数使用的内部滤波器分离出来，绘出其幅频响应。

9.14 利用函数upfirdn 和一个最优法设计得到的低通 FIR 滤波器重复习题9.12。列出滤波器系数，绘出滤波器的幅频响应。说明用过的所有假设。

9.15 利用函数 resample 重复习题9.14。

9.16 (a) 在框图层次上设计一个二级抽取滤波器，按因子30降低一个音频信号的抽样率，并满足以下指标：

输入抽样频率 F_s	240 kHz
数据中感兴趣的最高频率	3.4 kHz
通带波纹 δ_p	0.05
阻带波纹 δ_s	0.01

你可以假定滤波器是直接FIR形式，其长度由下式确定：

$$\text{滤波器长度，} N = \frac{-10\log(\delta_p\delta_s) - 13}{14.6\Delta f} + 1$$

其中

Δf = 归一化过渡带宽

假定1级和2级的抽取因子分别是15和2。

(b) 计算最优法得到的1级和2级的抗混叠滤波器系数，绘出其幅频响应。

(c) 利用合适的MATLAB函数和根据下面连续信号得到的离散时间信号，检验抽取滤波器：

$$x(t) = 2\sin(2\pi 100t) + 3\sin(2\pi 1000t) + 2\sin(2\pi 3400t)$$

参考文献

Adams R.W. (1986) Design and implementation of an audio 18 bit analog-to-digital converter using oversampling techniques. *J. Audio Engineering Society*, **34**(3), 153–66.

Agrawal B.P. and Shenoi K. (1983) Digital methodology for ΣΔM. *IEEE Trans. Communications*, **31**(3), 360–70.

Claasen T.A.C.M., Mecklenbrauker W.F.G., Peek J.B.H. and Van Hurck N. (1980) Signal processing method for improving the dynamic range of A/D and D/A converters. *IEEE Trans. Acoustics, Speech and Signal Processing*. **28**(5), 529–38.

Crochiere R.E. and Rabiner L.R. (1975) Optimum FIR digital filter implementations for decimation, interpolation, and narrow-band filtering. *IEEE Trans. Acoustics, Speech and Signal Processing*, **23**(5), 444–56.

Crochiere R.E. and Rabiner L.R. (1976) Further considerations in the design of decimators and interpolators. *IEEE Trans. Acoustics, Speech and Signal Processing*, **24**, 296–311.

Crochiere R.E. and Rabiner L.R. (1979) A program for multistage decimation, interpolation, and narrow band filtering. In *IEEE Programs for DSP*. Institute of Electrical and Electronics Engineers.

Crochiere R.E. and Rabiner L.R. (1981) Interpolation and decimation of digital signals – a tutorial review. *Proc. IEEE*, **69**(3), 300–31.

Crochiere R.E. and Rabiner L.R. (1983) *Multirate Digital Signal Processing*. Englewood Cliffs NJ: Prentice-Hall.

Crochiere R.E. and Rabiner L.R. (1988) Multirate processing of digital signals. In *Advanced Topics in Signal Processing* (Lim J.S. and Oppenheim A.V. (eds)). Englewood Cliffs NJ: Prentice-Hall.

DeFatta D.J., Lucas J.G. and Hodgkiss W.S. (1988) *Digital Signal Processing: A System Design Approach*. New York: Wiley.

Liu B. and Mintzer F. (1978) Calculation of narrow-band spectra by direct decimation. *IEEE Trans. Acoustics, Speech and Signal Processing*, **26**(6), 529–34.

Matsuya Y., Uchimura K., Iwata A., Kobayashi T., Ishikawa M. and Yoshitome T. (1987) A 16-bit oversampling A-to-D conversion technology using triple integration noise shaping. *IEEE J. Solid State Circuits*, **22**(6), 921–8.

Quarmby D. (ed.) (1984) *Signal Processor Chips*, Chapter 5. London: Granada.

Welland D.R., Del Signore B.P., Swanson E.J., Tanaka T., Hamashita, K., Hara S. and Takasuka K. (1989) A stereo 16-bit delta-sigma A/D converter for digital audio. *J. Audio Engineering Society*, **37**(6), 476–86.

Westgate J.A., Keith R.D.F., Gurnow J.S.K., Ifeachor E.C. and Greene K.R.G. (1990) Suitability of fetal scalp electrodes for fetal electrocardiogram during labour. *J. Clin. Physics & Physiological Measurement*, **11**(4), 297–306.

参考书目

Analog Devices (1988) *ADSP-2100 Family Applications Handbook*, Volume 2, Chapter 3. Analog Devices, Inc.

Bellanger M.G. (1977) Computation rate and storage estimation in multirate digital filtering with halfband filters. *IEEE Trans. Acoustics, Speech and Signal Processing*, **25**, 344–6.

Bellanger M.G., Daquet J.L. and Lepagnol G.P. (1974) Interpolation, extrapolation and reduction of computation speed in digital filters. *IEEE Trans. Acoustics, Speech and Signal Processing*, **22**, 231–5.

Brown Jr J.L. (1981) Multichannel sampling of lowpass signals. *IEEE Trans. Circuits Systems*, **28**, 101–6.

Cox R.V., Bock D.E., Bauer K.B., Johnston J.D. and Snyder J.H. (1987) The analogue voice privacy system. *AT&T Technical J.*, **66**, 119–31.

Elliot D.F. (ed.) (1987) *Handbook of Digital Signal Processing*. New York: Academic Press.

Goedhart D., van der Plassche R.J. and Stikvoort E.F. (1982) Digital-to-analog conversion in playing a compact disc. *Philips Technical Rev.*, **40**(6), 174–9.

Goodman D.J. and Carey M.J. (1977) Nine digital filters for decimation and interpolation. *IEEE Trans. Acoustics, Speech and Signal Processing*, **25**(2), 121–6.

Goodman D.J. and Flanagan J.L. (1971) Direct digital conversion between linear and adaptive delta modulation formats. In *Proc. IEEE Int. Communications Conf.*, Montreal, Canada, June 1971.

Huber A., De Man E., Schiller E. and Ulbrich W. (1986) FIR lowpass filter for signal decimation with 15 MHz clock frequency. In *IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Tokyo, 7–11 April, pp. 1533–6.

Jerri A.J. (1977) The Shannon sampling theorem – its various extensions and applications: a tutorial review. *Proc. IEEE*, **65**(11), 1565–96.

Linden D.A. (1959) A discussion of sampling theorems. *Proc. IRE.*, **47**, 1219–26.

Mintzer F. (1982) On half-band, third-band and *N*th-band FIR filters and their design. *IEEE Trans. Acoustics, Speech and Signal Processing*, **30**, 734–8.

Mintzer F. and Liu B. (1978) The design of optimal multirate bandpass and bandstop filters. *IEEE Trans. Acoustics, Speech and Signal Processing*, **26**(6), 534–43.

Mintzer F. and Liu B. (1978) Aliasing error in the design of multirate filters. *IEEE Trans. Acoustics, Speech and Signal Processing*, **26**, 76–88.

Montijo B.A. (1988) Digital filtering in a high-speed digitizing oscilloscope. *Hewlett Packard J.*, June, 70–6.

Mou Z.J. and Duhamel P. (1987) Fast FIR filtering: algorithms and implementations. *Signal Processing*, **13**, 377–84.

Princen J.P. and Bradley A.B. (1986) Analysis/synthesis filter banks design based on time domain aliasing cancellation. *IEEE Trans. Acoustics, Speech and Signal Processing*, **23**, 1153–61.

Rabiner L.R. and Crochiere R.E. (1975) A novel implementation for narrow-band FIR digital filters. *IEEE Trans. Acoustics, Speech and Signal Processing*, **23**(5), 457–64.

Regalia P.A., Fujii N., Mitra S.K. and Neuvo Y. (1987) Active RC crossover networks with adjustable characteristics. *J. Audio Engineering Society*, January–February, **35**(1/2), 24–30.

Rorabacher D.W. (1975) Efficient FIR filter design for sample rate reduction and interpolation. In *Proc. IEEE International Symposium on Circuits and Systems*, 21–23 April, pp. 396–9.

Schafer R.W. and Rabiner R.L. (1973) A digital signal processing approach to interpolation. *Proc. IEEE*, **61**, 692–702.

Scheuermann H. and Gockler H. (1981) A comprehensive survey of digital transmultiplexing methods. *Proc. IEEE*, **69**, 1419–50.

Shannon C.E. (1949) Communications in the presence of noise. *Proc. IRE*, **37**, 10–21.

Thong T. (1989) Practical consideration for a continuous time digital spectrum analyzer. In *Proc. IEEE International Symposium on Circuits and Systems*, Portland OR, May 1989, 1047–50.

Tuffs D.W., Rorabacher D.W. and Mosier W.E. (1970) Designing simple, effective digital filters. *IEEE Trans. Audio Electro-Acoustics*, **18**, 142–58.

Vaidyanathan P.P. (1990) Multirate digital filters, filter banks, polyphase networks, and applications: a tutorial. *Proc. IEEE*, **78**(1), 56–93.

Vaidyanathan P.P. (1993) *Multirate Systems and Filter Banks*. Englewood Cliffs NJ: Prentice-Hall.

Vaidyanathan P.P. and Nguyen T.Q. (1987) A trick for the design of FIR half-band filters. *IEEE Trans. Circuits and Systems*, **34**, 297–300.

Van De Plassche R.J. and Dijkmans E.C. (1983) A monolithic 16-bit D/A conversion system for digital audio. In *Digital Audio* (Blesser B. (ed.)), pp. 54–60. Audio Engineering, Inc.

Zobel R.N. and Tang P.S. (1985) A high performance multichannel decimating FIR digital filter system for microprocessor based data acquisition. *Proc. ISCAS*, 1149–52.

附录

9A C 语言程序—多抽样率处理和系统设计

下面的 C 语言程序可在本书指导手册的 CD 中找到（参见前言），其中还包含演示例子。

(1) decimate.c，使用最高到三级的抽取处理函数；

(2) interpol.c，使用最高到三级的内插处理函数；

(3) moptimum.c，确定一个 I（$I = 1, 2, 3, 4$）级抽取滤波器（或内插滤波器）的特性。特性包括抽取因子和各级滤波器特性，各种参数下的有效性分析（如每秒乘法次数）。

9B MATLAB ——多抽样率数字信号处理

利用MATLAB信号处理工具箱可以进行一系列有关抽样率变换和多抽样率处理的操作。本节我们着重讲解用于多抽样率处理的 MATLAB 函数和它们的作用。

就像正文中所说的，设计一个实用的多抽样率系统包括以下步骤：

(1) 确定抽样率变换器的指标；

(2) 确定抽样率变换器的参数（如级数、抽样率变换因子和各级的滤波器特性）；

(3) 设计抗混叠滤波器和 / 或去镜像滤波器；

(4) 多抽样率滤波，增加抽样率和 / 或降低抽样率。

MATLAB 信号处理工具箱完全能够实现步骤 3 和步骤 4。4 个用于多抽样率处理的 MATLAB 函数是 decimate、interp、resample 和 upfirdn。还有其他许多函数，特别是用于 FIR 滤波器的函数，共同组成了多抽样率 DSP 函数库。

decimate 函数用于按因子 M 降低一个数据序列的抽样率，其中 M 是一个正整数。函数使用一个低通滤波器进行抗混叠滤波，再把信号在低的抽样率上重新抽样，即每隔 M 个数据点只保留一个数据点。decimate 函数的执行句法为

```
y=decimate(x,M)
y=decimate(x,M,N)
y=decimate(x,M,'fir')
y=decimate(x,M,N,'fir')
```

y = decimate(x,M,N,'fir')执行先将矢量 x 表示的数据序列进行 N 点 FIR 低通滤波，其归一化截止频率为 $1/M$，再按因子 M 降低抽样率的操作。滤波器是由工具箱自动产生的。

如果参数 'fir' 被省略（例如上面的第一个句法），decimate 函数使用一个 n 阶切比雪夫 I 型滤波器，其归一化截止频率为 $0.8/M$，通带波纹为 0.05 dB（默认的 N 值是 8）。若使用 IIR 滤波器，函数同时应用前后时间方向滤波器来补偿相位失真。

interp 函数用于按整数因子 L 增加一个数据序列的抽样率。函数首先通过在数据抽样中加入零来扩展数据序列，再自动设计一个低通 FIR 滤波器进行去镜像滤波操作。其执行句法为

```
y=interp(x,L)
y=interp(x,L,N,alpha)
[y,b]=interp(x,L,N,alpha)
```

参数N决定了滤波器长度（滤波器长度$=2^{NL+1}$，默认值为4），alpha是归一化截止频率（默认值是0.5，即alpha $=f_c/f_{\text{Nyquist}}$）。最后的操作是将去镜像滤波器的系数输出到矢量b中。

例 9B.1 一个连续时间信号由下式给出

$$x(t)=A\cos(2\pi f_1 t)+B\cos(2\pi f_2 t)$$

(a) 利用MATLAB产生其等效的离散时间信号。假定抽样频率为1 kHz，$f_1=50$ Hz，$f_2=100$ Hz，且两个频率分量幅度比为$A/B=1.5$。

(b) 利用interp函数按因子4对离散时间信号进行内插。

(c) 利用decimate函数按因子4对步骤(b)内插滤波器的输出进行抽取。

(d) 绘出原始的、内插后的和抽取后的离散时间信号。

解：

MATLAB的m文件如程序9B.1所示。原始的、内插后的和抽取后的离散时间信号分别参见图9B.1(a)、图9B.1(b)和图9B.1(c)。读者应留意图9B.1(a)和图9B.1(c)的区别，它是由抽样率变换操作的不完善造成的。

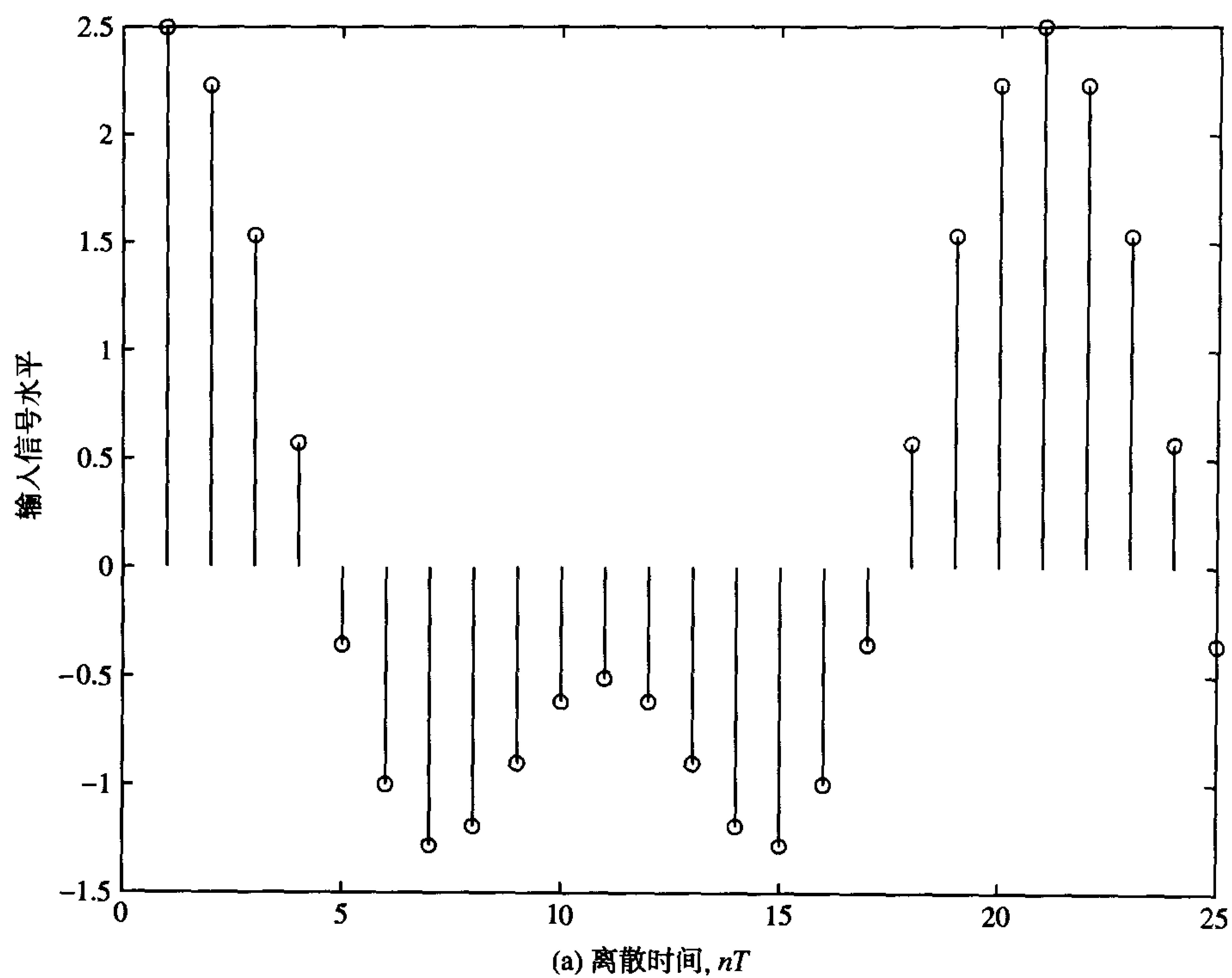
程序 9B.1　用于说明简单的内插和抽取操作的MATLAB的m文件

```
%
% File name: Program EX9B1.m
% An Illustration of interpolation by a factor of 4
%
Fs=1000;                              % sampling frequency
A=1.5;                                % relative amplitudes
B=1;
f1=50;                                % signal frequencies
f2=100;
t=0:1/Fs:1;                           % time vector
x=A*cos(2*pi*f1*t)+B*cos(2*pi*f2*t);  % generate signal
y=interp(x,4);                        % interpolate signal by 4
stem(x(1:25))                         % plot original signal
xlabel('Discrete time, nT')
ylabel('Input signal level')
figure
stem(y(1:100))                        % plot interpolated signal.
xlabel('Discrete time, 4 x nT')
ylabel('Interpolated output signal level')
y1=decimate(y,4);
stem(y1(1:25))                        % plot decimated signal.
xlabel('Discrete time, nT')
ylabel('Decimated output signal level')
```

decimate和interp函数用于快速执行抽样率变换。然而，它们具有某些局限（如抗混叠和去镜像滤波器是自动产生的）。upfirdn和resample函数则具有更多的设计选择性。

upfirdn和resample函数可以用于按有理数因子L/M（L和M是正整数）增加和/或降低一个数据序列的抽样率。两个命令进行类似的操作。upfirdn的执行句法为

```
y=upfirdn(x, h)
y=upfirdn(x, h, L)
y=upfirdn(x, h, L, M)
```



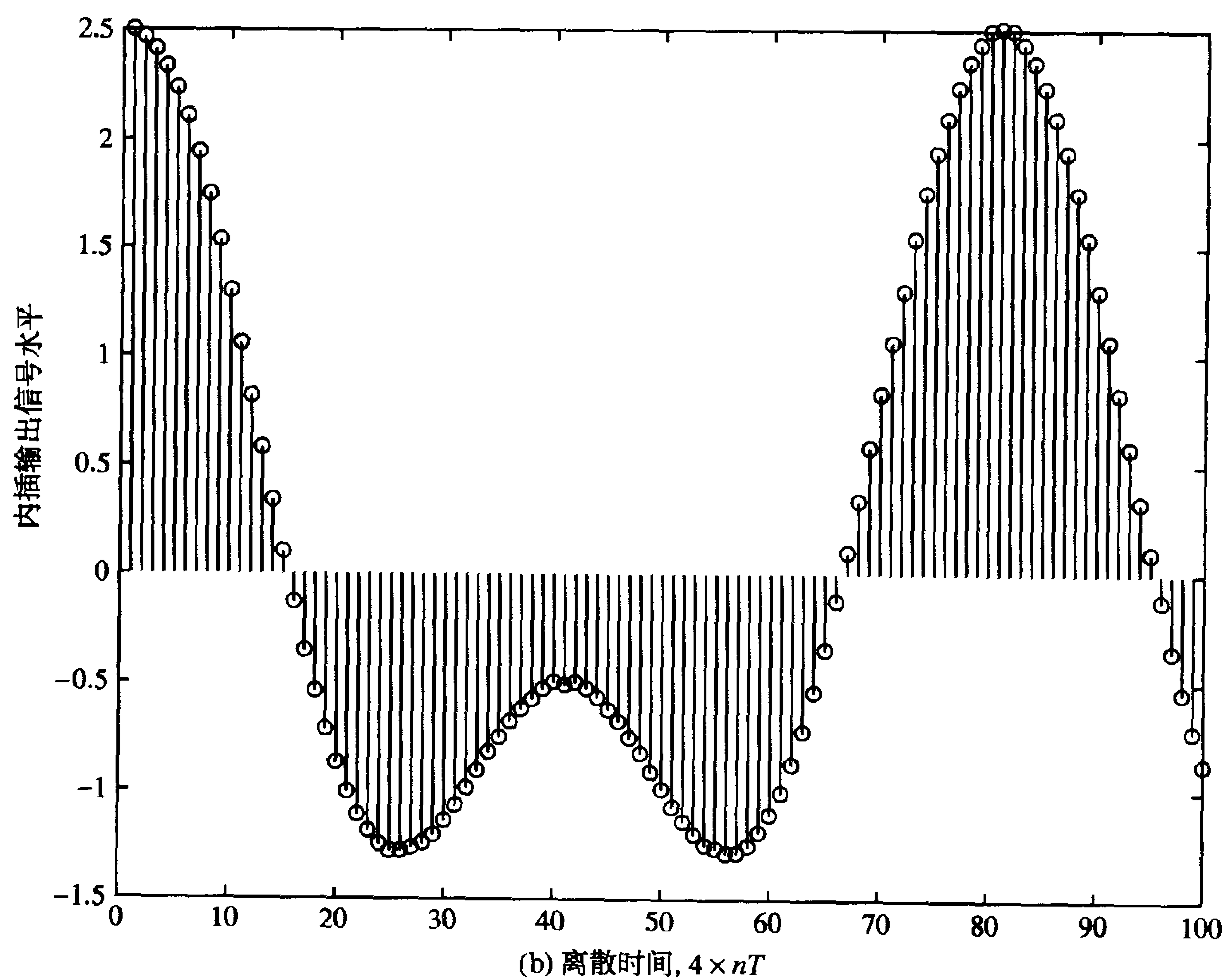


图 9B.1　(a)输入信号；(b)按因子 4 的内插；(c)按因子 4 的抽取

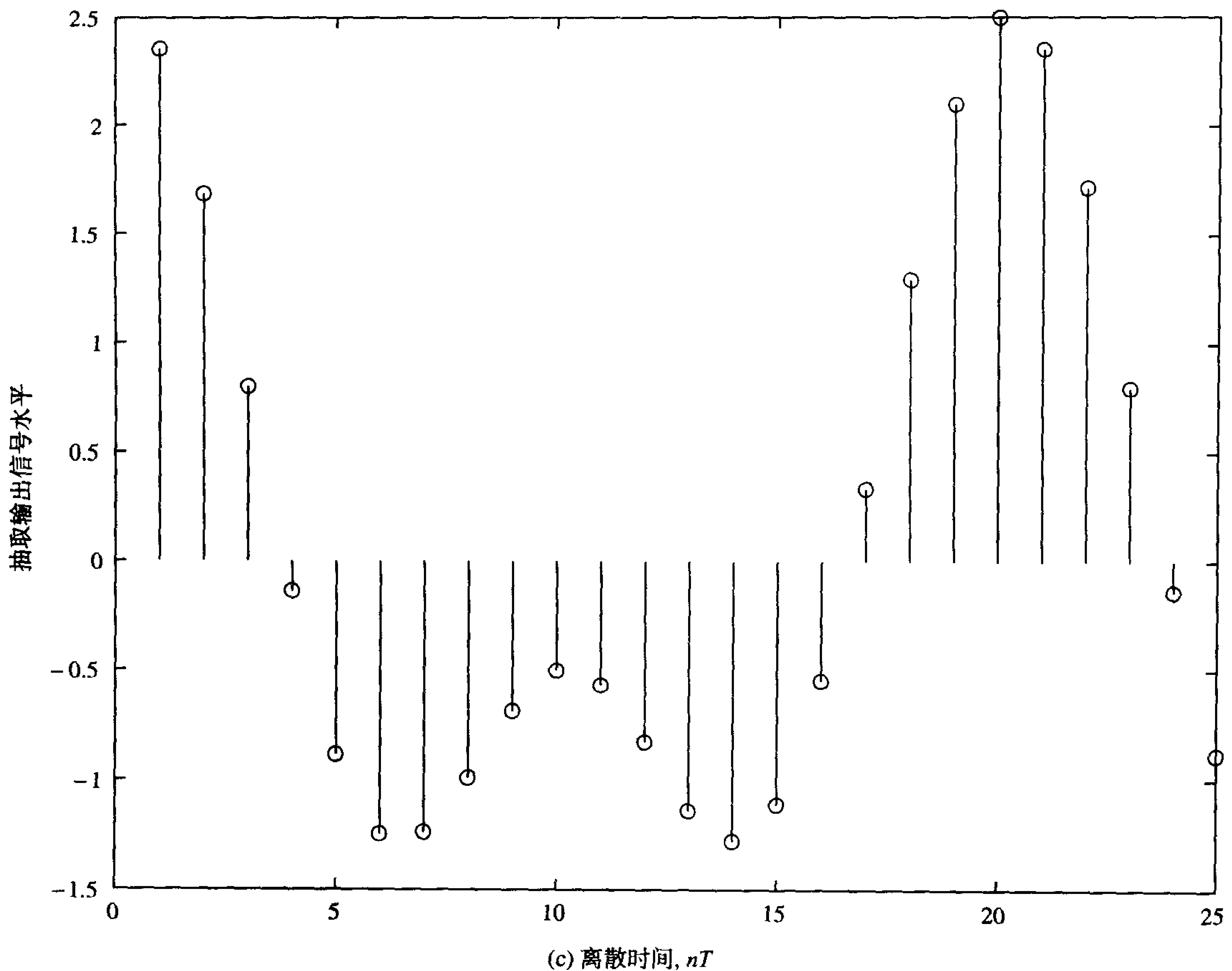


图9B.1（续） (a) 输入信号；(b) 按因子4的内插；(c) 按因子4的抽取

以 y = upfirdn(x, h, L, M)为例，首先按因子 L 增加抽样率，使用矢量 b 中的滤波器系数执行FIR滤波操作，再按因子 M 降低序列的抽样率。可以用最优、加窗或频率抽样的方法来设计FIR滤波器。执行返回的输出信号存于矢量 y 中。如果 L 和 M 都等于1，则仅执行一个FIR滤波操作。如果 $L = 1$，数据按因子 M 被抽取（或降低其抽样率）；若 $M = 1$，数据按因子 L 被内插（或增加其抽样率）。

resample 的执行句法为

```
y=resample(x, L, M)
[y, b]=resample(x, L, M)
y=resample(x, L, M, b)
```

y = resample(x, L, M)执行带限矢量 x 表示的数据序列的操作，其滤波使用凯塞窗和fir1语句设计的FIR滤波器。在上面第二个语句[y, b] = resample(x, L, M)中，可以获得FIR的滤波器系数。y = resample (x, L, M, b)则允许用户使用自己设计的FIR滤波器。

作为一个说明，我们使用resample来解决最后的问题（参见例9B.1），即按因子4增加和降低抽样率。MATLAB m文件请参见程序9B.2。由于执行的差异，它们的结果很相似但不完全相同。

程序 9B.2　用于说明简单的内插和抽取操作的 MATLAB m 文件

```
%
%   File name: EX9B2.m
%   An Illustration of sampling rate changes using resample by a factor of 4
%
Fs=1000;                              % sampling frequency
A=1.5;                                % relative amplitudes
B=1;
f1=50;                                % signal frequencies
f2=100;
t=0:1/Fs:1;                           % time vector
x=A*cos(2*pi*f1*t)+B*cos(2*pi*f2*t);  % generate signal
y=resample(x,4,1);                    % interpolate signal by 4
stem(x(1:25))                         % plot original signal
xlabel('Discrete time, nT')
ylabel('Input signal level')
figure
stem(y(1:100))                        % plot interpolated signal.
xlabel('Discrete time, 4 x nT')
ylabel('Interpolated output signal level')
y1=resample(y,1,4);
figure
stem(y1(1:25))                        % plot decimated signal.
xlabel('Discrete time, nT')
ylabel('Decimated output signal level')
```

第10章　自适应数字滤波器

一个自适应滤波器实际上是一个可以自动调整其特性的数字滤波器。它根据输入信号自动地变化。自适应滤波器是DSP的子领域——自适应信号处理的中心课题。本章基于最小均方（LMS）和递归最小平方（RLS）两种在自适应信号处理中最广泛应用的算法来介绍这一课题的重点方面。基于实际的处理，在正文中只给出了基本理论。在指导手册的CD（参见前言）中，包含了一系列基于LMS和RLS的自适应滤波器的C语言实现程序，并提供了大量真实世界的应用例子。

10.1　何时使用自适应滤波器及应用的范围

在实际应用中，经常出现感兴趣信号被其他不需要的、通常较强的信号或噪声污染的问题。如果信号或噪声占据了固定和分隔的频带，一般使用常规的固定系数线性滤波器来滤出信号。然而，还有许多情况是滤波器特性必须能够根据信号特性的改变而改变，或者说智能改变。在这种情形下，滤波器系数是变化的而不能事先设定。现有一个信号与噪声频谱重叠的例子（参见图10.1），噪声占据的频带是未知或时变的。固定系数滤波器不适用的典型应用有以下一些方面：

(1) 脑电图（EEG）。人工干扰如眼睛运动和眨眼产生的干扰信号远大于大脑本身的电活动，且二者占据了同样的频带。常规的线性滤波器不可能在滤出人工干扰的同时保留对医疗有用的信号。

(2) 宽频数字通信。强的阻塞信号可能干扰希望的信号，中断通信。这种干扰通常占据宽带频谱中一个未知的窄带，只能自适应地处理。

(3) 通过电话信道进行高速数字数据通信。该信道的幅度及相位响应特性较差，信号失真导致代表不同信号码的脉冲互相干扰（码间干扰），使接收端难以可靠地确定码字。为了补偿接收端的信道失真，它是时变或未知的，需要自适应均衡。

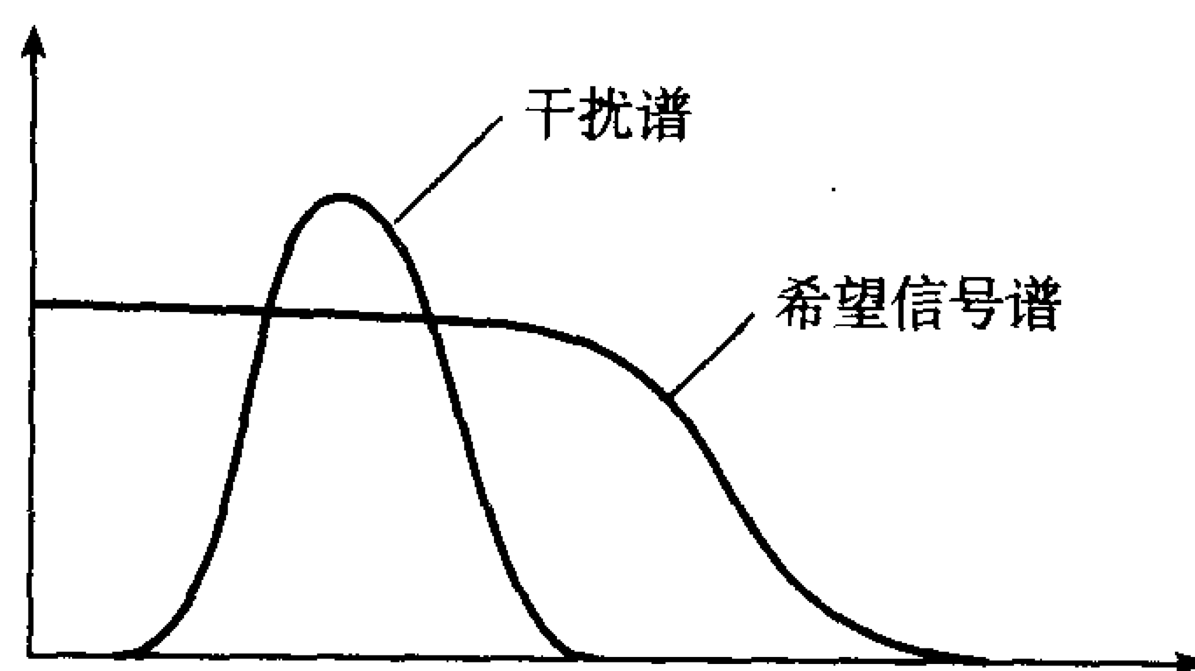


图10.1　一个信号和一个强干扰频谱重叠的示意图

一个自适应滤波器具有频率响应是可调或按照某些准则自动改变以提高其性能的性质，使滤波器根据输入信号的特性而改变。由于它们的自调整性能和内在的可塑性，自适应滤波器被应用于许多不同的领域，如电话回声对消、雷达信号处理、导航系统、通信信道均衡和生物医学信号增强等。

使用自适应滤波器的情况总结为

- 当需要滤波器特性变化以适应改变的情况时；
- 当信号和噪声存在频谱重叠（参见图 10.1）时；
- 或者噪声占据的频谱是时变或未知时。

在上述例子中应用常规滤波器会导致希望信号难以接受的失真。另外还有许多不属于噪声消除的其他情况，但同样适用于自适应滤波器（参见后文）。

10.2　自适应滤波的概念

10.2.1　自适应滤波器作为噪声对消器

一个自适应滤波器包括两个不同的部分：一个具有可调系数的数字滤波器，以及一个用于调整或改变滤波器系数的自适应算法（参见图 10.2）。两个输入信号 y_k 和 x_k，被同时加在自适应滤波器上。信号 y_k 是被污染信号，包含了所希望的信号 s_k 和噪声 n_k，且假设二者是互相不相关的。信号 x_k 是被污染信号的某种测量，与 n_k 具有相关性。x_k 被数字滤波器处理以产生一个噪声 n_k 的估计 $\hat{n}_k$。希望信号的估计可以从被污染信号 y_k 中减去数字滤波器输出的噪声估计 $\hat{n}_k$ 而获得：

$$\hat{s}_k = y_k - \hat{n}_k = s_k + n_k - \hat{n}_k \tag{10.1}$$

噪声消除的主要目的是产生被污染信号中噪声的最优估计，这样就获得了信号的最优估计。它是通过一种合适的自适应算法，在反馈网络中利用 $\hat{s}_k$ 来调整数字滤波器系数，从而最大程度地降低 $\hat{s}_k$ 中的噪声分量。输出信号 $\hat{s}_k$ 具有两个作用：(i) 希望信号的一个估计；(ii) 用于调整滤波器系数的误差信号。

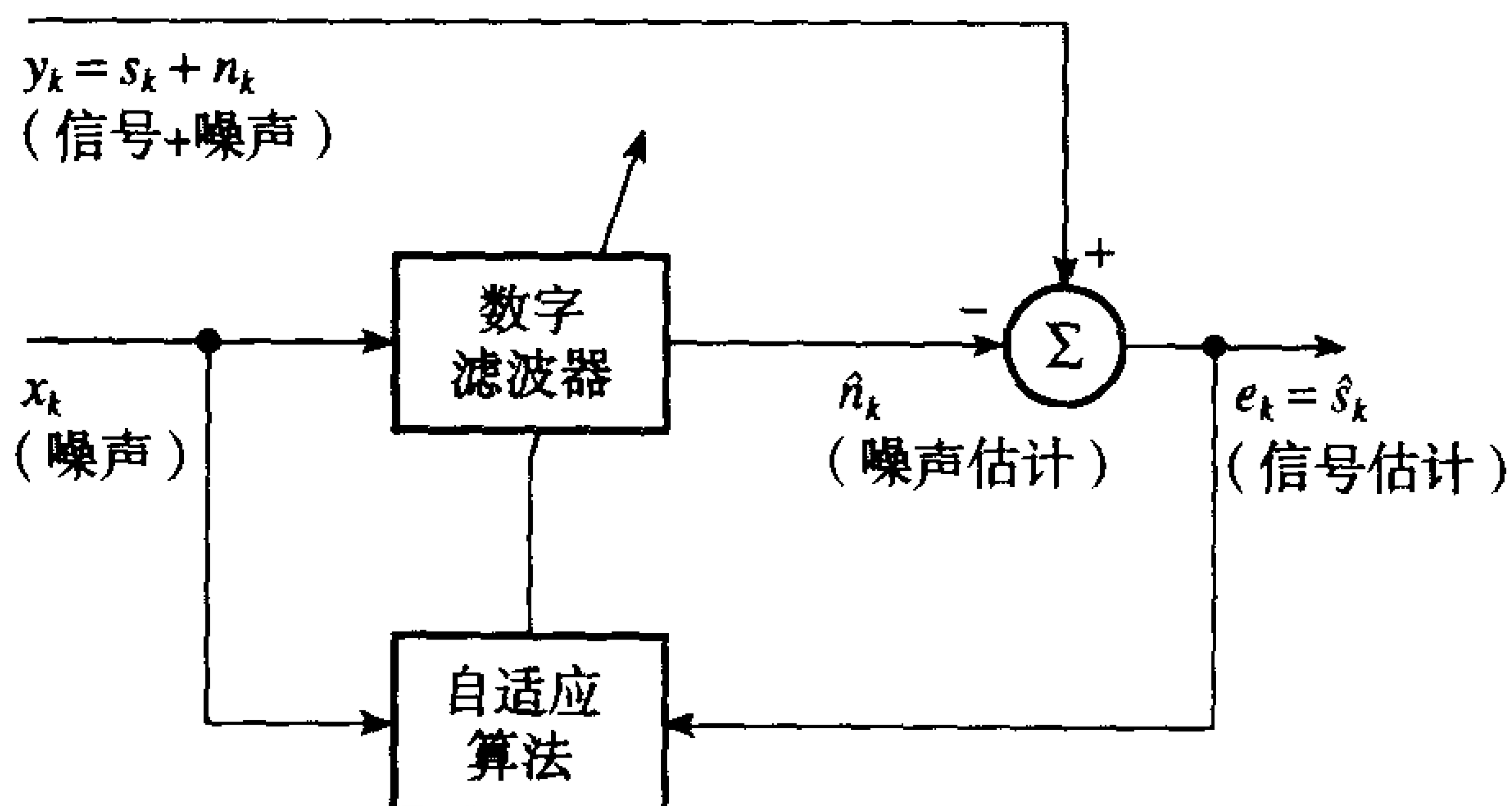


图 10.2　自适应滤波器作为噪声对消器的框图

10.2.2　自适应滤波器的其他形式

上述讨论是基于自适应噪声对消器的原理。需要重视的是自适应滤波器还用于许多其他的目的，如线性预测、自适应信号增强和自适应控制等。一般来说，信号 x_k、y_k 和 e_k 的含义或得到它们的方法都是取决于应用场合的，这是必须牢记的事实。图 10.3 给出了不同应用形式的自适应滤波器。

10.2.3　自适应滤波器的主要部件

在大多数自适应系统中，图 10.2 的数字滤波器是通过一个横向或有限冲激响应（FIR）滤波器结构（参见图 10.4）来实现的。其他结构有时也用到，如无限冲激响应（IIR）或格形结构，但 FIR 结构由于其简洁性和可靠的稳定性是最广泛应用的。对于图 10.4 描述的 N 点滤波器，输出为

$$\hat{n}_k = \sum_{i=0}^{N-1} w_k(i) x_{k-i} \tag{10.2}$$

其中 $w_k(i), i=0, 1,\ldots$ 是可调的滤波器系数（或权重），$x_k(i)$ 和 $\hat{n}_k$ 分别是滤波器的输入和输出。图 10.4 给出了一个单输入单输出系统。在多输入单输出系统中，x_k 可能是从 N 个不同信号源同时输入的。

(a) 自适应噪声对消器

(b) 自校准滤波器

(c) 没有外部参考源时对消周期性干扰

(d) 自适应增强线

(e) 系统建模

(f) 线性组合器

图 10.3 自适应滤波器的一些不同形式（Widrow and Winter, 1988）

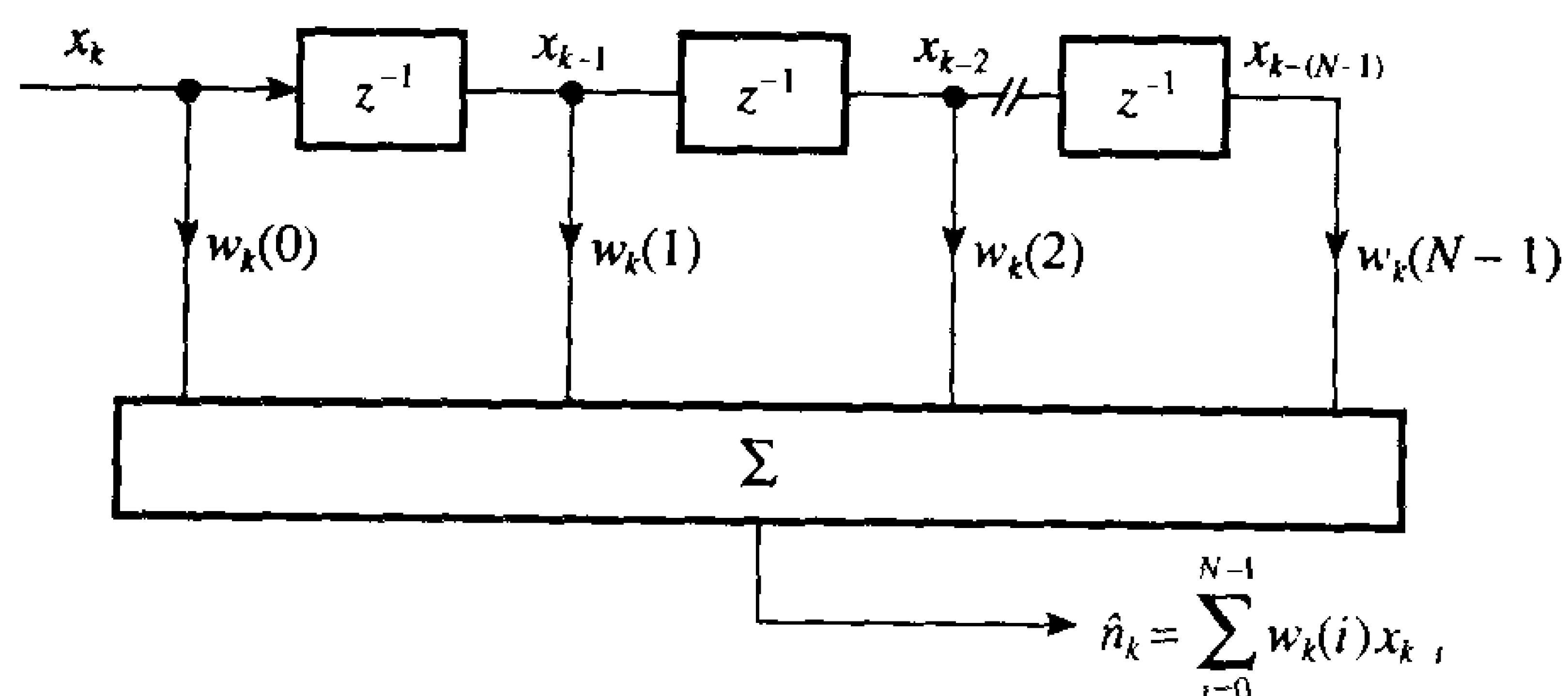


图 10.4 有限冲激响应滤波器结构

10.2.4　自适应算法

自适应算法用于调整数字滤波器的系数（参见图 10.2），使误差信号 e_k 根据某种准则，如最小均方准则（LMS），而被最大程度地降低。广泛应用的算法有最小均方、递归最小平方和卡尔曼（Kalman）滤波算法。考虑到计算和存储量需求，LMS 算法是最有效的。另外，它还不会遇到其他两种算法固有的数值不稳定问题。由于这些原因，LMS 算法成为许多应用场合的首选。然而，RLS 算法具有更优越的收敛性质。

例 10.1　在自适应噪声对消器的输出端，对希望信号的估计为（Widrow et al., 1975a）

$$\hat{s}_k = y_k - \hat{n}_k = s_k + n_k - \hat{n}_k$$

可以看出，最小化对消器的输出总功率就能最大化输出信噪比。

解：

被污染信号为

$$y_k = s_k + n_k \tag{10.3}$$

希望信号的估计为

$$\hat{s}_k = y_k - \hat{n}_k = s_k + n_k - \hat{n}_k \tag{10.4}$$

对 10.4 式两边开平方得

$$\hat{s}_k^2 = s_k^2 + (n_k - \hat{n}_k)^2 + 2s_k(n_k - \hat{n}_k) \tag{10.5}$$

对 10.5 式两边取数学期望得

$$E[\hat{s}_k^2] = E[s_k^2] + E[(n_k - \hat{n}_k)^2] + 2E[s_k(n_k - \hat{n}_k)] \tag{10.6}$$

因为所希望信号 s_k 与 n_k 和 $\hat{n}_k$ 均不相关，10.6 式的最后一项为 0，有

$$E[\hat{s}_k^2] = E[s_k^2] + E[(n_k - \hat{n}_k)^2] \tag{10.7}$$

其中，$E[s_k^2]$ 代表总信号功率，$E[\hat{s}_k^2]$ 代表估计信号的功率（也代表输出总功率），$E[(n_k - \hat{n}_k)^2]$ 代表 s_k 中的剩余噪声功率。显然，10.7 式中如果估计 $\hat{n}_k$ 是 n_k 的完全复制，则输出功率就只包含信号的功率。通过调节自适应滤波器到最佳位置，剩余噪声功率和总输出功率被最小化。由于 s_k 与 n_k 无关，希望信号功率则不受影响。即

$$\min E[\hat{s}_k^2] = E[s_k^2] + \min E[(n_k - \hat{n}_k)^2] \tag{10.8}$$

很清楚，在 10.8 式中，最小化输出信号功率的真正效果是最大化了输出信噪比。当滤波器调节到使 $\hat{n}_k = n_k$，就有 $\hat{s}_k = s_k$。在本例中，自适应噪声对消器的输出则应是无噪的。当信号 y_k 不包含噪声时，即 $n_k = 0$，自适应滤波器自动关闭（至少理论上如此）或调节所有的权重为零。

10.3　基本维纳滤波器理论

许多自适应滤波算法都能被近似地看成是离散维纳滤波器（参见图 10.5）。两个信号 x_k 和 y_k 同时加在滤波器上。典型的 y_k 包含一个与 x_k 相关的分量和另一个与 x_k 不相关的分量。维纳滤波器则产生 y_k 中与 x_k 相关分量的最优估计，再从 y_k 中减去它就得到 e_k。

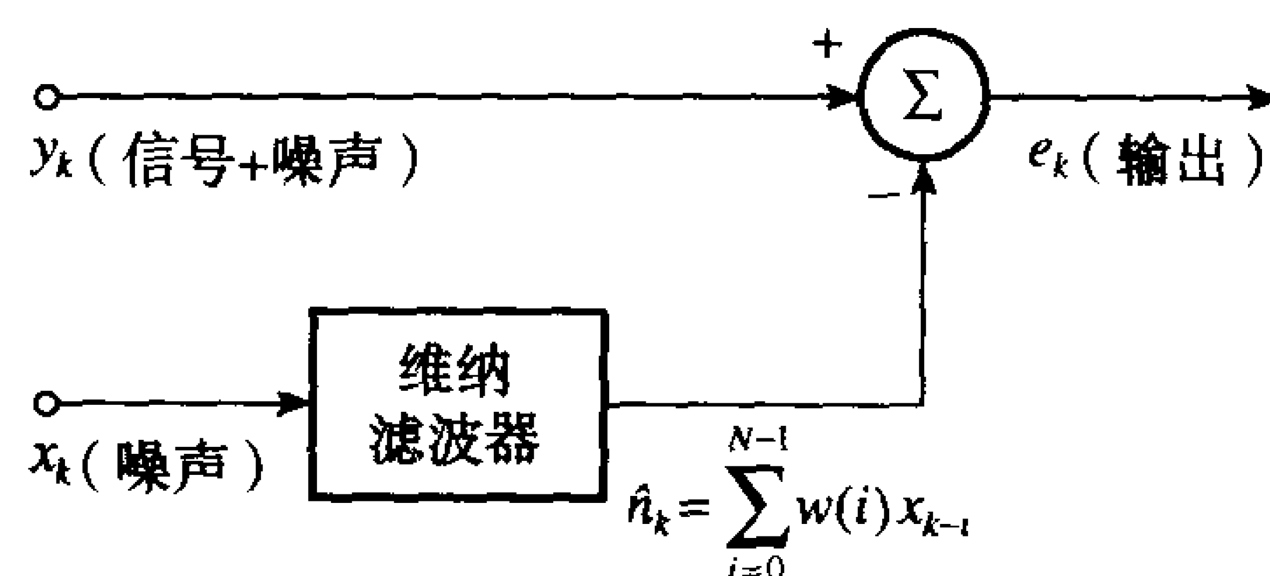


图 10.5 基本维纳滤波器

假定一个N个系数（或权重——文献中常用的提法）的FIR滤波器结构，维纳滤波器输出和原始信号y_k间的差信号e_k为

$$e_k = y_k - \hat{n}_k = y_k - \mathbf{W}^{\mathrm{T}}\mathbf{X}_k = y_k - \sum_{i=0}^{N-1} w(i)x_{k-i} \tag{10.9}$$

其中$\mathbf{X}_k$和$\mathbf{W}$分别是输入信号矢量和权矢量。由下式确定：

$$\mathbf{X}_k = \begin{bmatrix} x_k \\ x_{k-1} \\ \vdots \\ x_{k-(N-1)} \end{bmatrix} \qquad \mathbf{W} = \begin{bmatrix} w(0) \\ w(1) \\ \vdots \\ w(N-1) \end{bmatrix} \tag{10.10}$$

误差平方为

$$e_k^2 = y_k^2 - 2y_k\mathbf{X}_k^{\mathrm{T}}\mathbf{W} + \mathbf{W}^{\mathrm{T}}\mathbf{X}_k\mathbf{X}_k^{\mathrm{T}}\mathbf{W} \tag{10.11}$$

将10.11式两边取期望得到均方误差（MSE）J，若输入矢量$\mathbf{X}_k$和信号y_k是联合平稳的：

$$\begin{aligned} J &= E[e_k^2] = E[y_k^2] - 2E[y_k\mathbf{X}_k^{\mathrm{T}}\mathbf{W}] + E[\mathbf{W}^{\mathrm{T}}\mathbf{X}_k\mathbf{X}_k^{\mathrm{T}}\mathbf{W}] \\ &= \sigma^2 + 2\mathbf{P}^{\mathrm{T}}\mathbf{W} + \mathbf{W}^{\mathrm{T}}\mathbf{R}\mathbf{W} \end{aligned} \tag{10.12}$$

其中$E[\cdot]$代表数学期望，$\sigma^2=E[y_k^2]$是y_k的方差，$\mathbf{P}=E[y_k\mathbf{X}_k]$是长度为$N$的互相关矢量，$\mathbf{R}=E[\mathbf{X}_k\mathbf{X}_k^{\mathrm{T}}]$是$N\times N$的自相关矩阵。一个MSE——滤波器系数的图形是碗形的，且只有惟一的底部（参见图10.6）。这个图形称为性能曲面，它是非负的。性能曲面的梯度由下式给出：

$$\nabla = \frac{\mathrm{d}J}{\mathrm{d}\mathbf{W}} = -2\mathbf{P} + 2\mathbf{R}\mathbf{W} \tag{10.13}$$

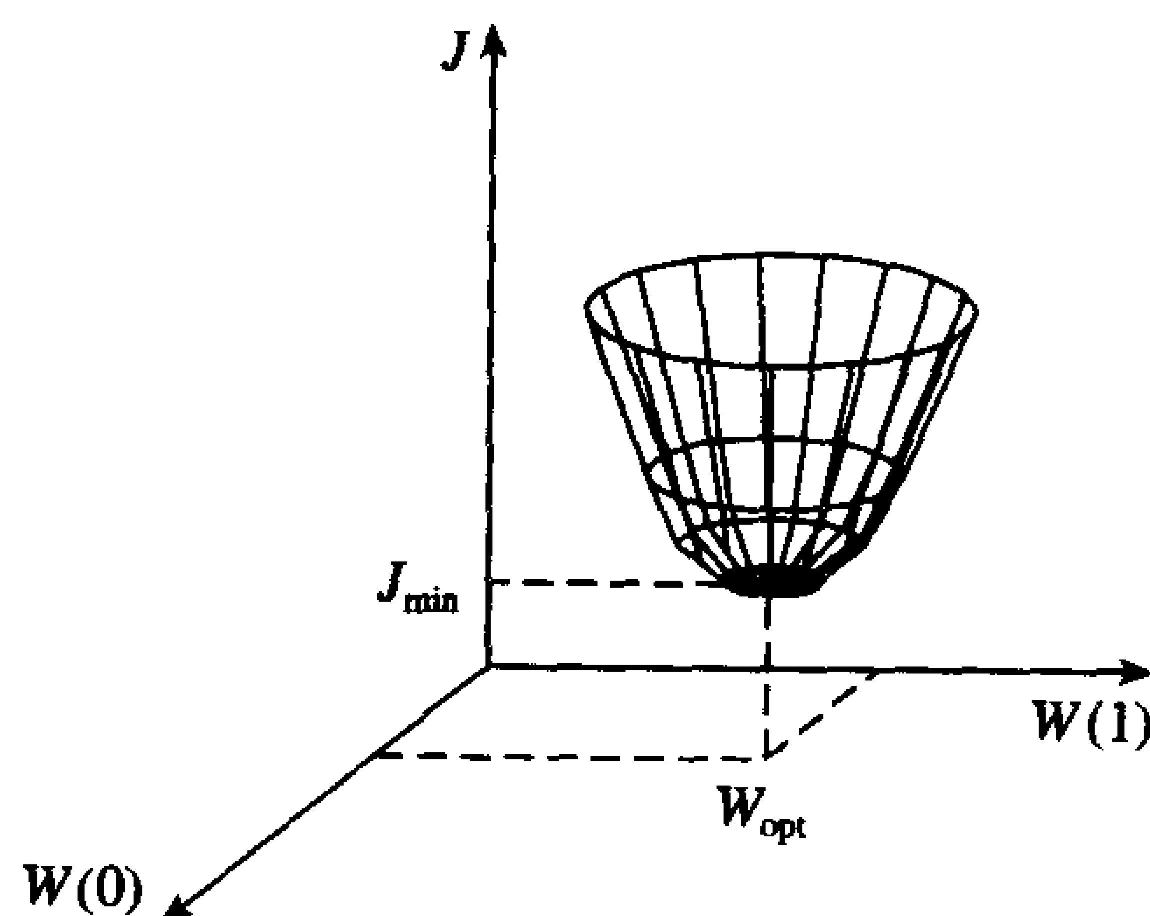


图 10.6 误差性能曲面

每组系数$w(i)$（$i=0, 1,\ldots, N-1$），对应了曲面上的一个点。在曲面上的最小点，梯度为零，滤波器权矢量到达最优值$\mathbf{W}_{\mathrm{opt}}$（参见图10.2）：

$$\mathbf{W}_{\text{opt}} = \mathbf{R}^{-1}\mathbf{P} \tag{10.14}$$

10.14式即著名的维纳－霍夫（Wiener-Hopf）方程或解。自适应滤波的任务是采用合适的算法来调节滤波器权重 $w(0), w(1),\ldots$，从而找到性能曲面的最优点。

维纳滤波器的实际用途是有限的，因为：

- 它需要已知自相关矩阵 $\mathbf{R}$ 和互相关矢量 $\mathbf{P}$，这两个量在事先通常是未知的；
- 它包含了矩阵求逆运算，非常耗费时间；
- 如果信号是非平稳的，则 $\mathbf{P}$ 和 $\mathbf{R}$ 都是时变的，导致必须重复计算 $\mathbf{W}_{\text{opt}}$。

对于实时应用，需要一种能够从依次加入的抽样点得到 $\mathbf{W}_{\text{opt}}$ 的算法。自适应算法就是用于达到这个目的，且不需要显式计算 $\mathbf{R}$ 和 $\mathbf{P}$ 或进行矩阵求逆。

例10.2　从均方误差方程（10.12式）出发，推导维纳－霍夫方程。

解：

MSE为

$$\text{MSE} = J = \sigma^2 + 2\mathbf{P}^{\text{T}}\mathbf{W} + \mathbf{W}^{\text{T}}\mathbf{R}\mathbf{W} \tag{10.15}$$

通过求MSE对权矢量 $\mathbf{W}$ 的偏微分，得到MSE的梯度 ∇，并令其为零（Haykin, 1986），则有

$$\nabla = \frac{\mathrm{d}J}{\mathrm{d}\mathbf{W}} = \frac{\mathrm{d}\sigma^2}{\mathrm{d}\mathbf{W}} + \frac{\mathrm{d}(\mathbf{P}^{\text{T}}\mathbf{W})}{\mathrm{d}\mathbf{W}} + \frac{\mathrm{d}(\mathbf{W}^{\text{T}}\mathbf{R}\mathbf{W})}{\mathrm{d}\mathbf{W}} \tag{10.16}$$

现在，

$$\frac{\mathrm{d}\sigma^2}{\mathrm{d}\mathbf{W}} = 0$$

$$\frac{\mathrm{d}(2\mathbf{P}^{\text{T}}\mathbf{W})}{\mathrm{d}\mathbf{W}} = -2\mathbf{P}$$

$$\frac{\mathrm{d}(\mathbf{W}^{\text{T}}\mathbf{R}\mathbf{W})}{\mathrm{d}\mathbf{W}} = 2\mathbf{R}\mathbf{W}$$

利用这些结果及 $\nabla = \mathbf{0}$，10.16式变成

$$\nabla = \frac{\mathrm{d}J}{\mathrm{d}\mathbf{W}} = -2\mathbf{P} + 2\mathbf{R}\mathbf{W} = \mathbf{0} \tag{10.17}$$

最优系数矢量由下式给出

$$\mathbf{W}_{\text{opt}} = \mathbf{R}^{-1}\mathbf{P} \tag{10.18}$$

10.4　基本LMS自适应算法

一个最成功的自适应算法是由Widrow及其同事（Widrow et al., 1975a）提出的LMS算法。不用像10.18式那样计算 $\mathbf{W}_{\text{opt}}$，在LMS算法中是根据依次加入的抽样点来调节系数以达到最小化MSE，即沿着图10.6的曲面下降至其底部。

LMS是基于最速下降的算法，其权矢量根据依次加入的抽样点更新：

$$\mathbf{W}_{k+1} = \mathbf{W}_k - \mu\nabla_k \tag{10.19}$$

其中，$\mathbf{W}_k$ 和 ∇_k 分别是第 k 个抽样时刻的权重矢量和真实的梯度矢量。μ 控制收敛速度和稳定性。

由于∇_k是通过估值10.17式得到的，10.19式的最速下降法仍旧需要知道**R**和**P**。LMS是一种能实际应用的算法，不需要进行10.18式的矩阵求逆或直接计算自相关和互相关系数就能得到滤波器权重$\mathbf{W}_k$的估计。Widrow-Hopf的LMS算法对权重的依次更新为

$$\mathbf{W}_{k+1}=\mathbf{W}_k+2\mu e_k\mathbf{X}_k \tag{10.20a}$$

其中

$$e_k=y_k-\mathbf{W}_k^{\mathrm{T}}\mathbf{X}_k \tag{10.20b}$$

显然，上面的LMS算法不需要事先已知信号的统计量（即相关性**R**和**P**），而使用它们的瞬时估计代替（参见例10.3）。LMS算法获得的权重只是一个估计值，但随着调节权重，这些估计值逐渐提高，滤波器也越来越适应信号特性。最终，权重值收敛。收敛条件为

$$0<\mu>1/\lambda_{\max} \tag{10.21}$$

其中，$\lambda_{\max}$是输入数据方差矩阵的最大特征值。在实际应用中，$\mathbf{W}_k$不会达到理论的最优点（维纳解），但会在其周围波动（参见图10.7）。

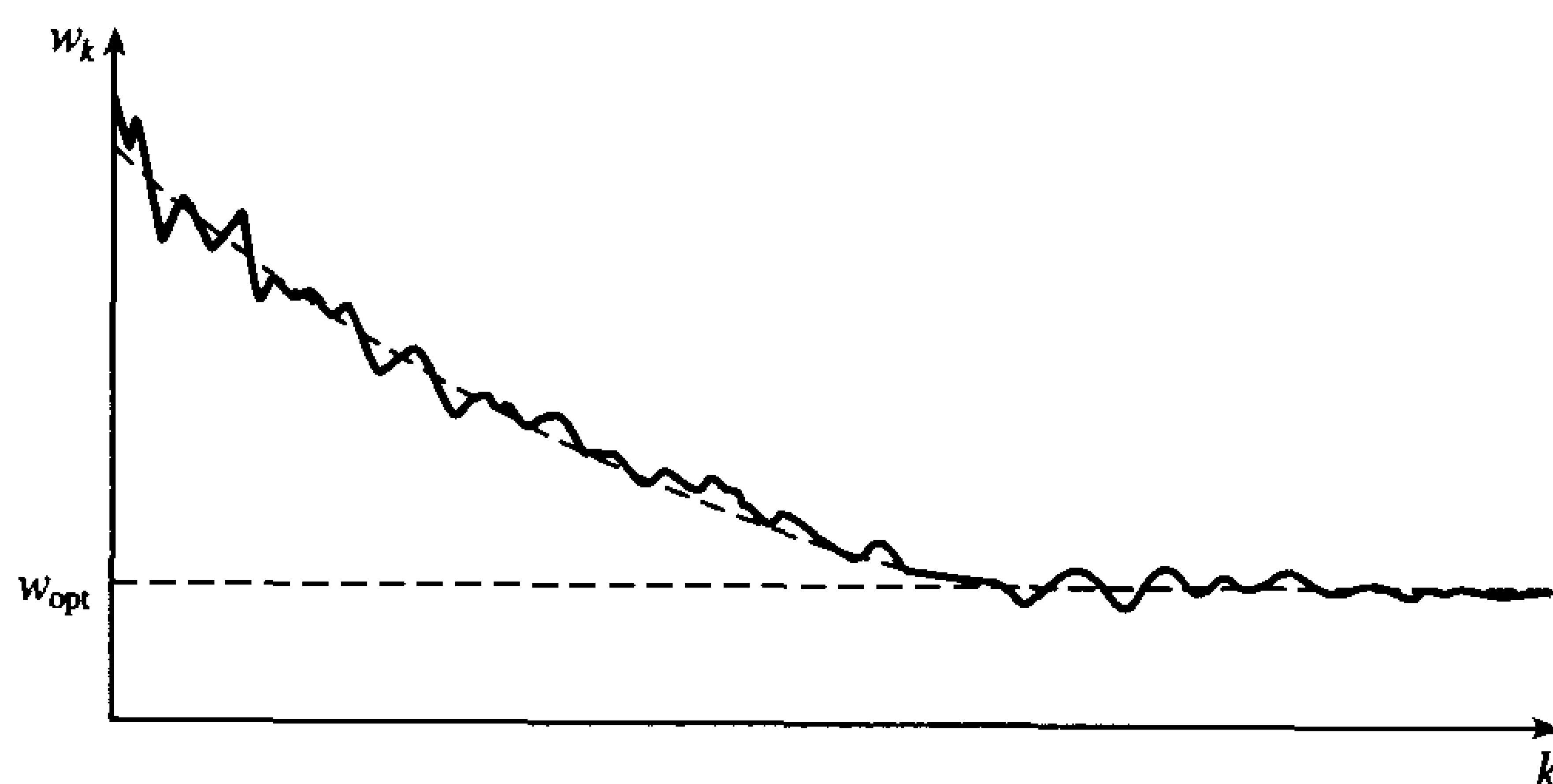


图 10.7 滤波器权重的变化示意图

10.4.1 基本LMS算法的实现

LMS算法的计算步骤总结如下。

(1) 初始化，令所有权重$w_k(i)$（$i=0, 1,..., N-1$）为任一固定值，或为零。

对每个接下来的抽样时刻（$k=1, 2, ...$），执行下面的步骤(2)到步骤(4)：

(2) 计算滤波器输出

$$\hat{n}_k=\sum_{i=0}^{N-1}w_k(i)x_{k-i}$$

(3) 计算误差估计

$$e_k=y_k-\hat{n}_k$$

(4) 更新下一时刻的滤波器权重

$$w_{k+1}(i)=w_k(i)+2\mu e_k x_{k-i}$$

从上面看出，LMS算法具有简洁和易于实现的特点，使它成为许多实时系统的算法首选。LMS算法对每组输入和输出抽样，大约需要2*N*+1次乘法和2*N*+1次加法。大多数信号处理器都适宜进行乘法－累加的算术操作，使直接实现LMS算法更具吸引力。

LMS算法的流图在图10.8中给出。图10.9和图10.10分别给出了软件实现的伪代码和硬件实现框图。一个C语言的LMS算法实现程序请参见附录。

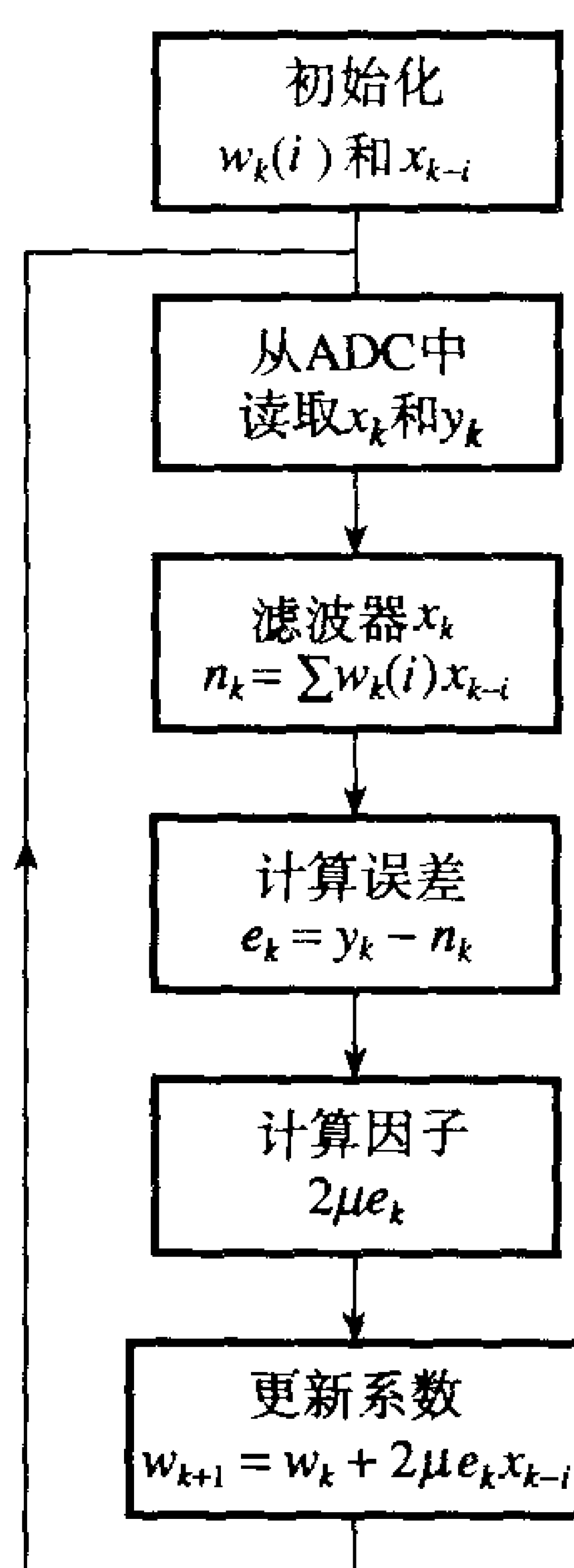


图 10.8　LMS 自适应滤波器的流程图

```
输入:          xk(i) 最近输入的抽样点矢量
               yk   当前被污染信号抽样点
               wk(i) 滤波器系数矢量
输出:          ek   当前希望(误差)的输出抽样点
               wk(i) 更新滤波器系数矢量

/* compute the current error estimate */

    ek=yk
    for i=1 to N do
      ek=ek-xk(i)*wk(i)
    end

/* update filter coefficients */

    gk=2u*ek
    for i=1 to N do
      wk(i)=wk(i)+xk(i)*gk
    end

    return
```

图 10.9　LMS 自适应滤波器程序

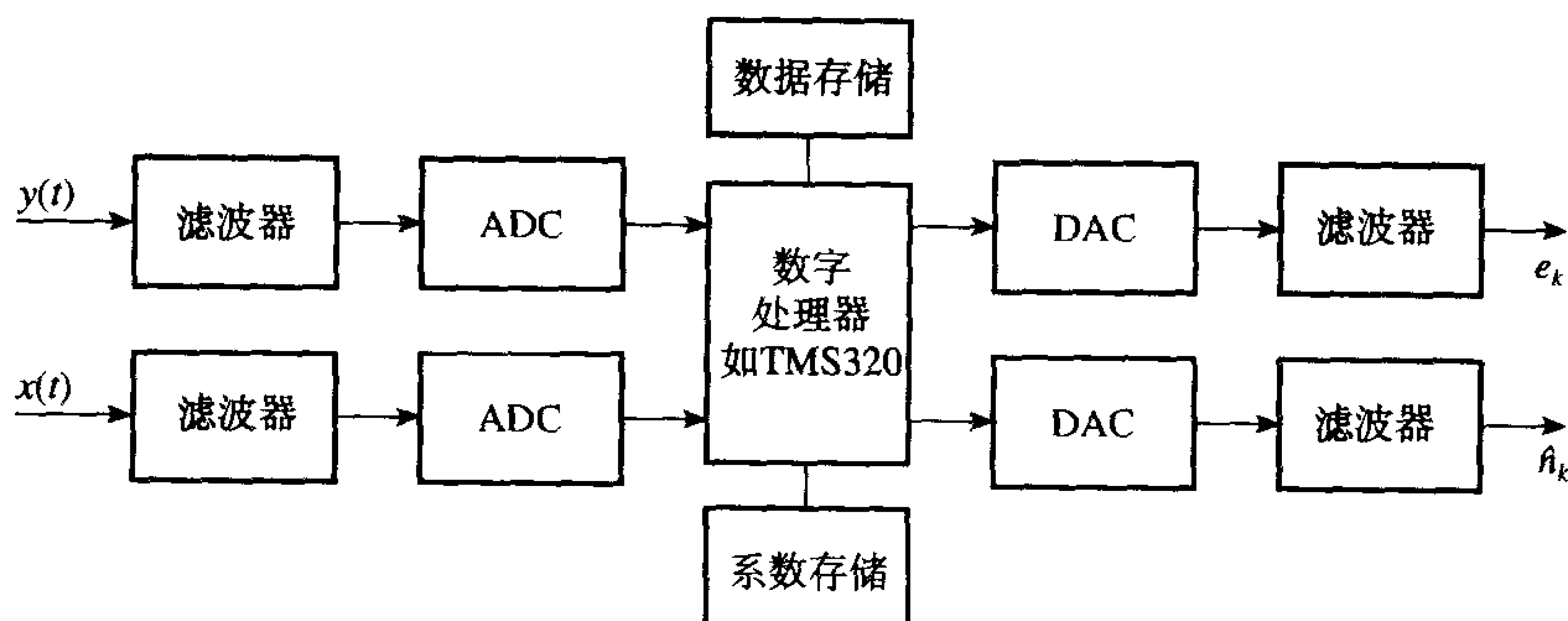


图 10.10　实时 LMS 自适应滤波的硬件实现

例 10.3 从最速下降法出发：

$$\mathbf{W}_{k+1} = \mathbf{W}_k - \mu\nabla_k$$

其中，$\mathbf{W}_k$是第k个抽样时刻的滤波器权矢量，μ控制收敛稳定性和速率，∇_k是误差－性能曲面的真实梯度，推导自适应噪声消除的 Widrow-Hopf 的 LMS 算法，说明任何用到的合理假设。

解：

最速下降法由下式给出：

$$\mathbf{W}_{k+1} = \mathbf{W}_k - \mu\nabla_k \tag{10.22}$$

梯度矢量∇、初级输入与次级输入的互相关$\mathbf{P}$以及初级输入的自相关$\mathbf{R}$之间的关系为

$$\nabla = -2\mathbf{P} + 2\mathbf{R}\mathbf{W} \tag{10.23}$$

在 LMS 算法中，使用∇的瞬时估计，则有

$$\begin{aligned}\nabla_k &= -2\mathbf{P}_k + 2\mathbf{R}_k\mathbf{W}_k = -2\mathbf{X}_k y_k + 2\mathbf{X}_k\mathbf{X}_k^{\mathrm{T}}\mathbf{W}_k \\ &= -2\mathbf{X}_k(y_k - \mathbf{X}_k^{\mathrm{T}}\mathbf{W}_k) = -2e_k\mathbf{X}_k\end{aligned} \tag{10.24}$$

其中

$$e_k = y_k - \mathbf{X}_k^{\mathrm{T}}\mathbf{W}_k$$

用 10.24 式替换最速下降法的梯度，我们得到基本 Widrow-Hopf 的 LMS 算法：

$$\mathbf{W}_{k+1} = \mathbf{W}_k + 2\mu e_k\mathbf{X}_k \tag{10.25a}$$

其中

$$e_k = y_k - \mathbf{W}_k^{\mathrm{T}}\mathbf{X}_k \tag{10.25b}$$

10.4.2 基本 LMS 算法的实际限制

在应用基本LMS算法中，会遇到有一些实际问题，导致性能的下降。这里讨论几个重点问题。

10.4.2.1 非平稳性的影响

在平稳环境中，滤波器的误差性能曲面具有恒定的形状和指向，自适应滤波器只会收敛和运行于最优点或其附近。如果权重收敛后信号统计量发生改变，滤波器重新将它的权重调节到一组新值来进行响应。这需要信号统计量在两次改变间的速度足够慢，滤波器才能及时收敛。然而在非平稳环境下，（性能曲面的）底部或最小点在持续移动中，它的指向和曲率也可能改变（参见图 10.11）。这种情况下算法不仅需要寻找曲面的最小点，还要完成对变化位置的跟踪，很大程度上降低了算法性能。注意，如果一个变量的统计量（如均值、方差、自相关）随时间而变化，则是非平稳的。这种变化来源于如偶发的短时干扰（参见图 10.12）或劣点数据，经常干扰滤波权重的收敛。

许多策略被提出以克服这个困难，但一般来说需要增加基本LMS算法的复杂度。其中一个例子是时间－排序自适应滤波器（Ferrara and Widrow, 1981）。

10.4.2.2 干扰输入信道中信号分量的影响

算法的性能依赖于以下假定：测量的干扰（噪声）信号$x_k(i)$，与真实干扰具有很强的相关性，而与希望信号的相关性很弱（理论上为零）。在大多数情况下，这个条件不满足。在一些实际应用中，除干扰或噪声外，干扰输入信道还包含了低电平的希望信号分量。这导致希望信号分量在某种程度上被消除，这种情况如图 10.13 所示。尽管如此，Widrow et al.(1975a)指出，在自适应噪声消除中，仍能使希望信号的信噪比有显著的提高，其代价是信号的少量失真。然而，如果x_k只包含信

号而未包含噪声分量，则y_k中的希望信号会完全被消除。我们在生物医学处理中的工作同样证实了这一点（Ifeachor et al., 1986）。

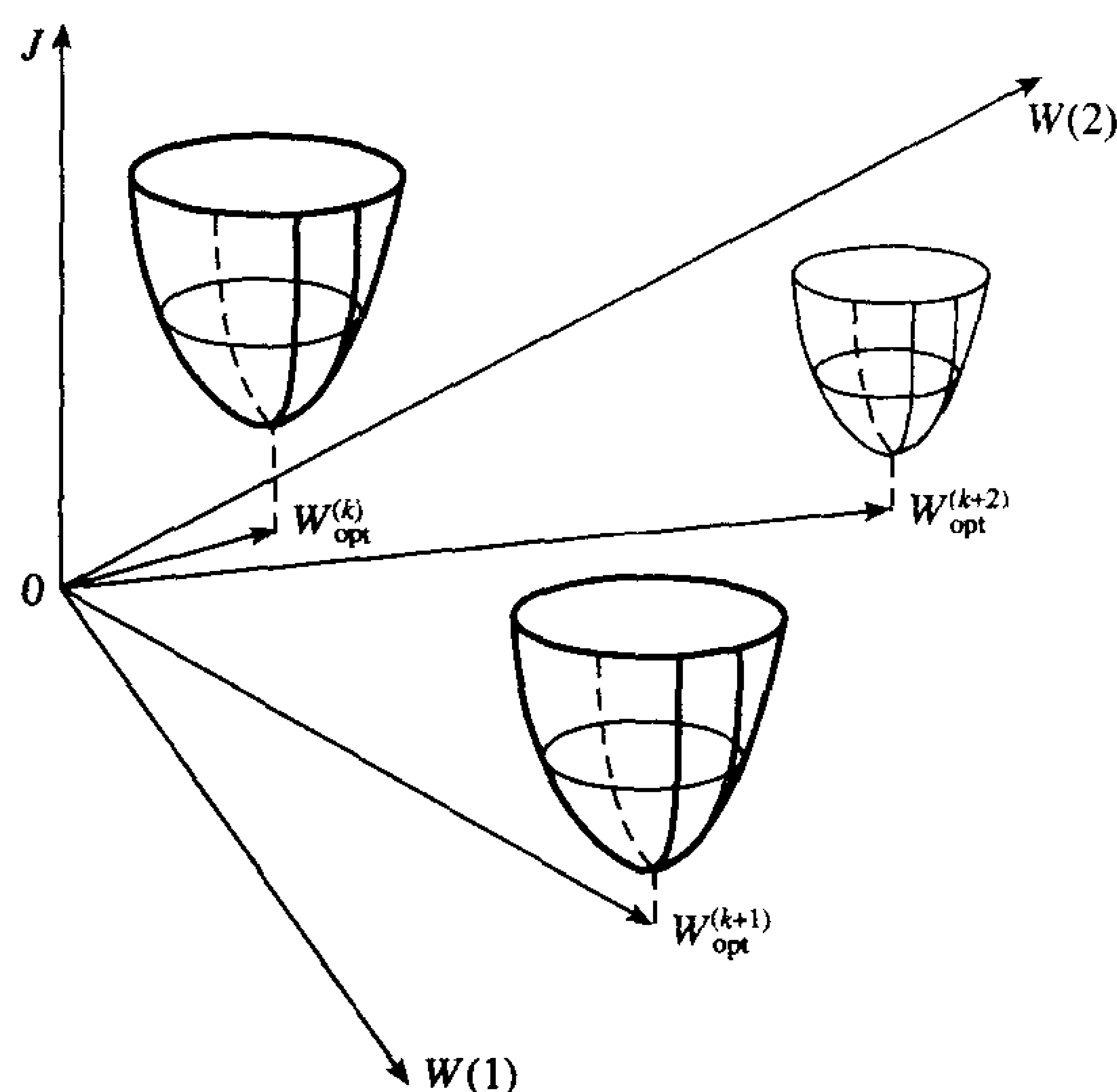
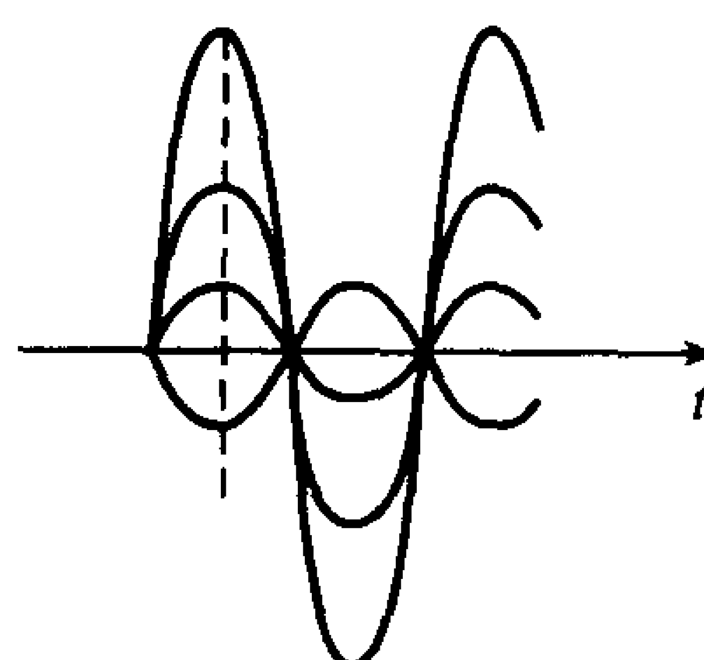
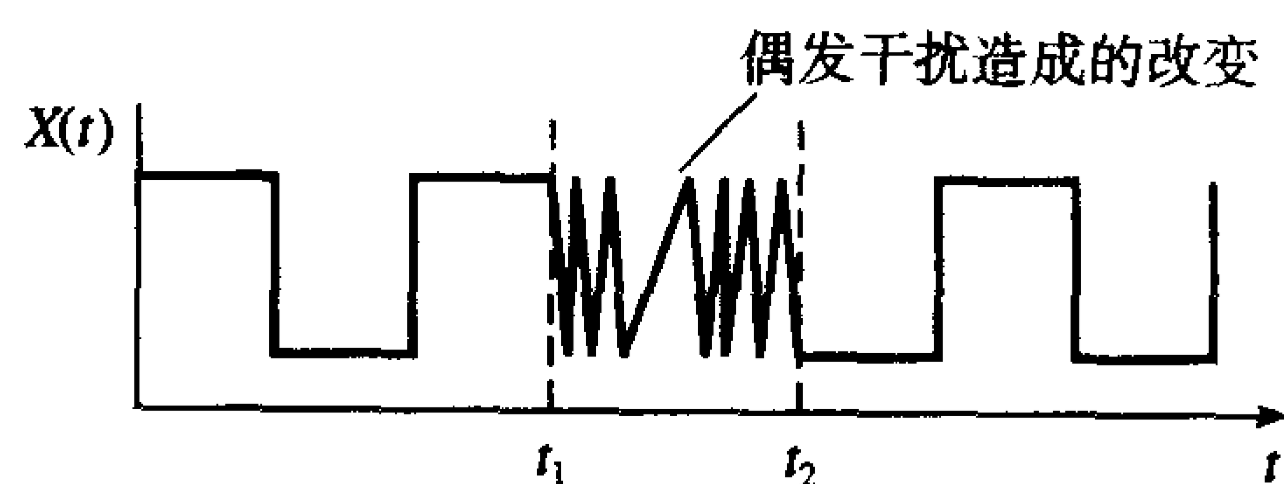


图 10.11 时变误差性能曲面



(a) 调制波形



(b) 偶发性干扰

图 10.12 非平稳过程的一种例子

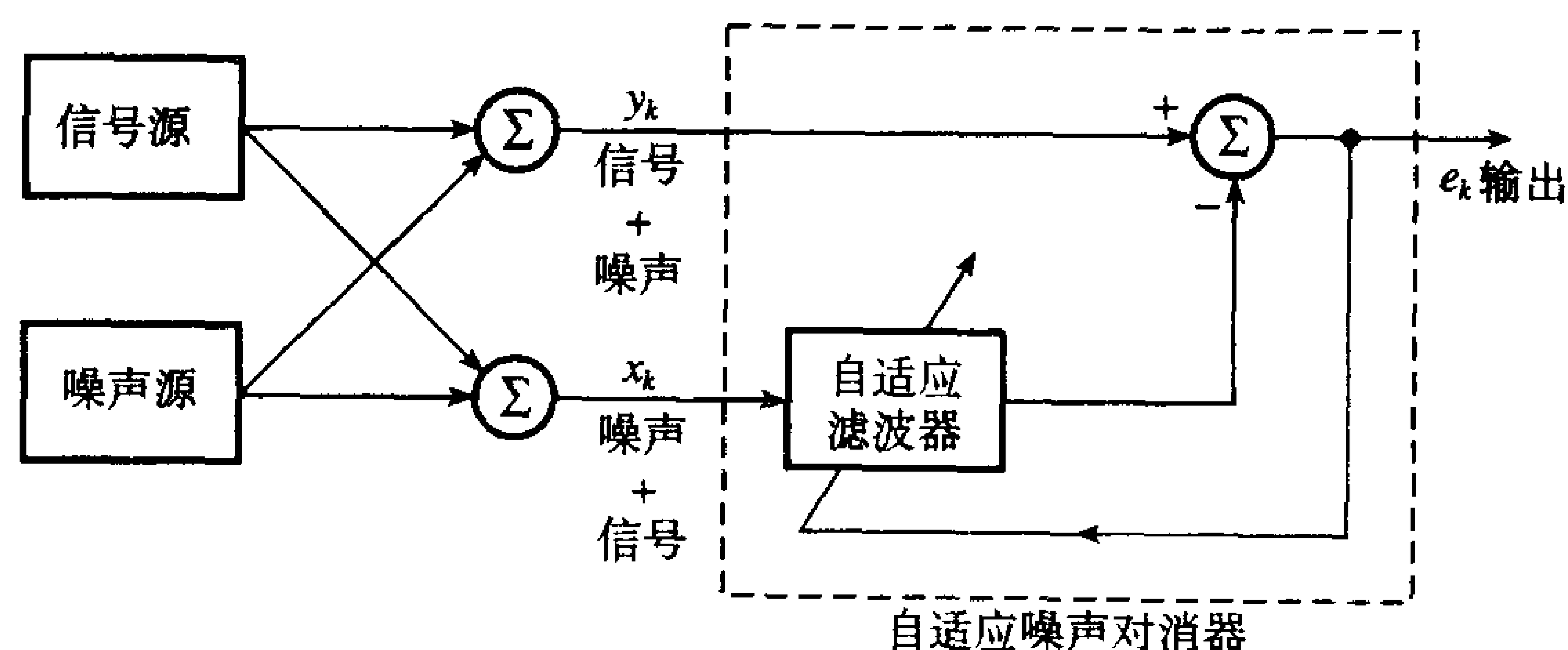


图 10.13 在希望信号输入与干扰输入的信道中，都含有一些信号分量情况下的自适应噪声消除

10.4.2.3 计算机字长要求

基于 LMS 的 FIR 自适应滤波器特性由以下方程确定：

$$\text{对数字滤波器} \quad \hat{n}_k = \sum_{i=0}^{N-1} w_k(i)x_{k-i} \tag{10.26a}$$

$$\text{对自适应算法} \quad \mathbf{W}_{k+1} = \mathbf{W}_k + 2\mu e_k \mathbf{X}_k \tag{10.26b}$$

其中

$$e_k = y_k - \mathbf{W}_k^{\mathrm{T}} \mathbf{X}_k$$

当自适应滤波器是在实际应用时，滤波器权重 w_k 和输入变量 x_k、y_k，都是用一定个数的比特位来表示的。同样，牵涉到的数字操作也具有一定的精度。LMS 算法的递归性质意味着字长效应会无限增长，所以一些比特必须在权重更新前被抛弃。由此 y_k、e_k 和 $w_k(i)$ 与它们的真实值有所不同。使用滤波器权重和有限精度算术操作会给自适应滤波器引入误差，其后果可能包括：(i) 自适应滤波器不能收敛到最优解，导致一个次等的性能（例如，滤波器用于干扰对消器，则会剩余一些干扰）；(ii) 滤波器输出还包含使其波动的噪声；(iii) 算法的提前终止。因此，必须保留足够的比特数以保证（有限字长）误差在许可范围之内。大多数公开文献描述的自适应系统用 8~16 位固定位数来表示数字信号 x_{k-i} 和 y_k，而系数则被量化成 16~24 位。乘法器的使用范围从 8 × 8 位到 24 × 16 位，累加器使用范围为 16~40 位。一般来说，对于低阶滤波器（系数最多到 100 个），使用不超过 16 位精度来存储系数、16 × 16 位乘法器和长度 32 位的累加器就足够了。

10.4.2.4 系数漂移

对于某些类型输入（如窄带信号），滤波器系数可能从最优值漂移开，慢慢地增长直到超出允许的字长范围。这是 LMS 固有的缺陷，导致一个长期的性能下降。在实际应用中，使用一个泄漏因子来解决系数漂移问题，即缓慢地推动系数向零的方向前进。这样的两种方式在 10.27 式中给出：

$$w_{k+1}(i) = \delta w_k(i) + 2\mu e_k x_{k-i} \qquad 0 < \delta < 1 \tag{10.27a}$$

$$w_{k+1}(i) = w_k(i) + 2\mu e_k x_{k-i} \pm \delta \qquad 0 < \delta < 1 \tag{10.27b}$$

小的泄漏因子 δ，保证了漂移是受控的，但会给 e_k 引入额外的偏差。

基本 LMS 算法的实用性被扩展到许多前面提到的更复杂的基于 LMS 的算法中。它们包括

(1) 复 LMS 算法，允许处理复数据；

(2) 分块 LMS 算法，可以给计算带来很大方便，有时能达到更快的收敛；

(3) 时序 LMS 算法，能处理某些类型的非平稳性。

10.4.3 其他基于 LMS 的算法

10.4.3.1 复 LMS 算法

复 LMS 算法更新滤波器权重的算法是（Widrow et al., 1975b）

$$\tilde{\mathbf{W}}_{k+1} = \tilde{\mathbf{W}}_k + 2\mu \tilde{e}_k \tilde{\mathbf{X}}_{k-i} \tag{10.28}$$

其中符号 ~ 代表一个复变量。Mitel PDSP16XXX 处理器适用于复 LMS 算法，因为它们能直接进行复数据的算术运算，比其他常规处理器具有明显的优势。

10.4.3.2　快速 LMS 算法

人们提出了许多种分块LMS算法，因其带来的计算量减小的优势，尤其是对于滤波器系数个数较大的情况。不用在一个时刻计算一个抽样，而是计算整个数据块。分块LMS算法利用了快速傅里叶变换（FFT）的计算优势，并在频域进行卷积过程（Mansour and Gray, 1982）。

一个有效的频域滤波器如图 10.14 所示。

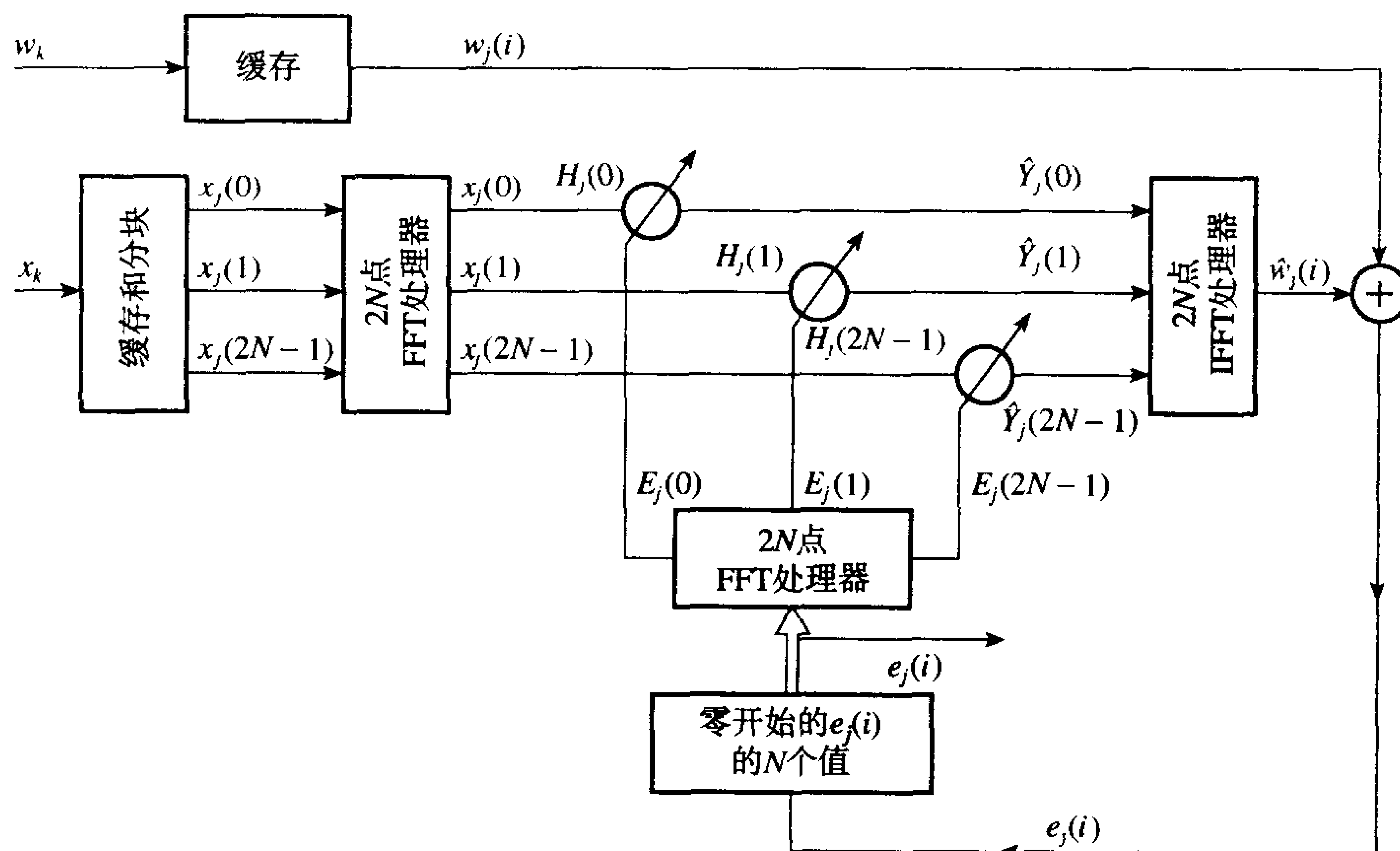


图 10.14　频域 LMS 滤波器的简化框图

10.5　递归最小二乘算法（RLS）

RLS算法是基于著名的最小二乘法（参见图10.15）。一个输出信号y_k，在离散时间k通过一组输入信号$x_k(i)$（$i = 1, 2,..., n$）的响应被观测。输入和输出信号间的关系为简单的递归模型：

$$y_k = \sum_{i=0}^{n-1} w(i)x_k(i) + e_k \qquad (10.29)$$

其中e_k代表观测误差或其他难以预计的效应，$w(i)$代表信号y_k中第i个输入的比（权）重。LS的问题是，给定$x_k(i)$和y_k，如何获得$w(0)$到$w(n-1)$的估计。

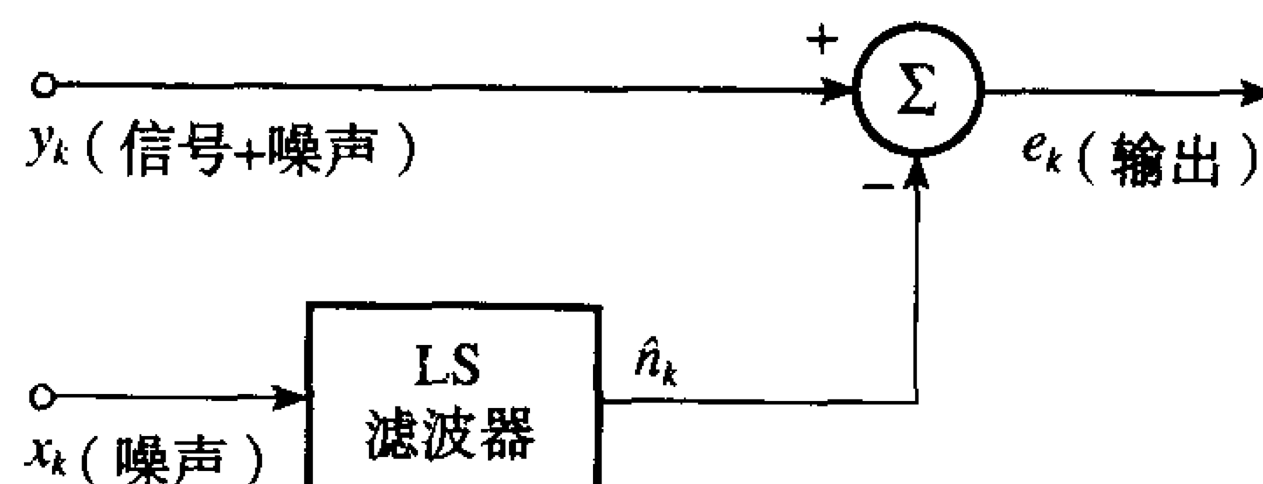


图 10.15　最小二乘方法的基本概念示意图

滤波器权重$w(i)$的最优估计（在最小平方的意义上）由下式给出：

$$\mathbf{W}_m = [\mathbf{X}_m^{\mathrm{T}}\mathbf{X}_m]^{-1}\mathbf{X}_m^{\mathrm{T}}\mathbf{Y}_m \qquad (10.30)$$

其中$\mathbf{Y}_m$、$\mathbf{W}_m$和$\mathbf{X}_m$为

$$\mathbf{Y}_m = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{m-1} \end{bmatrix} \qquad \mathbf{X}_m = \begin{bmatrix} \mathbf{x}^{\mathrm{T}}(0) \\ \mathbf{x}^{\mathrm{T}}(1) \\ \mathbf{x}^{\mathrm{T}}(2) \\ \vdots \\ \mathbf{x}^{\mathrm{T}}(m-1) \end{bmatrix} \qquad \mathbf{W}_m = \begin{bmatrix} w(0) \\ w(1) \\ w(2) \\ \vdots \\ w(n-1) \end{bmatrix}$$

$$\mathbf{x}^{\mathrm{T}}(k) = [x_k(0) \quad x_k(1) \quad \ldots \quad x_k(n-1)], \quad k = 0, 1, \ldots, m-1$$

下标 m 表明上面的每个矩阵是由 m 个数据点获得，T 表示转置。10.30 式给出了 $\mathbf{W}_m$ 的 OLS 估计，可以用任意适当的矩阵求逆技术来获得。滤波器输出则为

$$\hat{n}_k = \sum_{i=0}^{n-1} \hat{w}(i) x_{k-i}, \quad k = 1, 2, \ldots, m \tag{10.31}$$

10.5.1 递归最小二乘算法

10.30 式中 $\mathbf{W}_m$ 的计算需要费时的矩阵求逆。显然，LS 法不适于实时或在线滤波。在实用中，当我们需要获得连续数据或我们希望用新数据来提高对 $\mathbf{W}_m$ 的估计精度时，递归方法是一种好的选择。使用递归最小二乘算法，$\mathbf{W}_m$ 的估计可以用新的一组数据来更新，并且不用重复地进行费时的矩阵直接求逆。

一个适合的RLS算法可以通过对数据进行指数加权来逐渐消除老数据对 $\mathbf{W}_m$ 估计的影响，并跟踪信号的慢变化特征。即

$$\mathbf{W}_k = \mathbf{W}_{k-1} + \mathbf{G}_k e_k \tag{10.32a}$$

$$\mathbf{P}_k = \frac{1}{\gamma} [\mathbf{P}_{k-1} - \mathbf{G}_k \mathbf{x}^{\mathrm{T}}(k) \mathbf{P}_{k-1}] \tag{10.32b}$$

其中

$$\mathbf{G}_k = \frac{\mathbf{P}_{k-1} \mathbf{x}(k)}{\alpha_k}$$

$$e_k = y_k - \mathbf{x}^{\mathrm{T}}(k) \mathbf{W}_{k-1}$$

$$\alpha_k = \gamma + \mathbf{x}^{\mathrm{T}}(k) \mathbf{P}_{k-1} \mathbf{x}(k)$$

$\mathbf{P}_k$ 实际上在递归计算矩阵 $[\mathbf{X}_k^{\mathrm{T}} \mathbf{X}_k]^{-1}$ 的逆。

时间变量 k 强调一个关键事实，即数据在每个抽样点都获得。γ 即所谓的遗忘因子。当 $\gamma = 1$ 时这种加权方式退化成通常的LS方法。典型 γ 值为0.98到1之间。较小的值给最近的数据以更大的权重，使估计值变得上下波动。在每个抽样点给 $\mathbf{W}_k$ 值以明显贡献的先前抽样点个数称为渐进抽样长度（ASL），且

$$\sum_{k=0}^{\infty} \gamma^k = \frac{1}{1-\gamma} \tag{10.33}$$

这有效地定义了RLS滤波器的记忆长度。当 $\gamma=1$，即对应着通常的LS，滤波器需要无限长的记忆。

10.5.2 递归最小二乘算法的限制

RLS算法是非常有效的，对每个抽样点的运算次数与10.32式中矩阵 $\mathbf{W}_k$ 和 $\mathbf{P}_k$ 的固定维数相当。这是进行有效的实时滤波所必需的。然而，当直接应用RLS算法时，还会遇到两个主要问题。第

一个是所谓的“爆炸”，如果信号 $x_k(i)$ 长时间保持为零值，这时矩阵 $\mathbf{P}_k$ 的更新变成单纯除以 γ（小于 1 的值）或按指数增长：

$$\lim_{k\to\infty}\mathbf{P}_k=\lim_{k\to\infty}\left(\frac{\mathbf{P}_{k-1}}{\gamma_{k-1}}\right) \tag{10.34}$$

第二个问题是 RLS 算法对计算舍入误差具有敏感性，导致矩阵 $\mathbf{P}$ 逐渐负定而最终不稳定。要想成功地估计 $\mathbf{W}$，有必要使矩阵 $\mathbf{P}$ 保持半正定，即等效于在 LS 算法中要求矩阵 $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ 是可逆的。但是，由于 10.32b 式中的减法运算，$\mathbf{P}$ 的正定不能保证。这个问题在多参数模型中更趋于严重，特别是如果变量是线性相关或算法在一个具有有限字长的小系统上执行。当算法被迭代执行了很长时间后，10.32b 式括号中的两项非常接近，在一个有限字长系统上进行二者的减法很容易出现误差，导致出现了负定的 $\mathbf{P}_k$ 矩阵。

数值不稳定问题可以通过适当地因子分解矩阵 $\mathbf{P}$ 从而避免 10.32b 式中的相减项来解决。这类因子分解算法具有更好的条件数和相当于使用双浮点数 RLS 算法的精度。两种这样的算法是平方根和 UD 分解算法。从存储和计算角度上来看，UD 算法更有效且常见。实际上，UD 算法是平方根算法的不含平方根的表达形式，因此具有与后者相同的性质。

10.5.3　因子分解算法

10.5.3.1　平方根算法

在平方根算法中，矩阵 $\mathbf{P}_k$ 被分解成（Peterka, 1975）

$$\mathbf{P}_k=\mathbf{S}_k\mathbf{S}_k^{\mathrm{T}} \tag{10.35}$$

其中 $\mathbf{S}_k$ 是一个上三角矩阵，$\mathbf{S}_k^{\mathrm{T}}$ 是其转置，构成了 $\mathbf{P}_k$ 的平方根。这样用 $\mathbf{S}_k$ 替代 $\mathbf{P}_k$ 来进行迭代更新，由于平方根的乘积永远是正的，$\mathbf{P}_k$ 的正定性就可以保证。$\mathbf{S}_k$ 的更新为

$$\mathbf{S}_k=\frac{1}{\gamma^{1/2}}\mathbf{S}_{k-1}\mathbf{H}_{k-1} \tag{10.36}$$

这里 $\mathbf{H}_k$ 是一个上三角矩阵。

10.5.3.2　UD 分解算法

在 UD 算法中 $\mathbf{P}_k$ 被分解为（Bierman, 1976）

$$\mathbf{P}_k=\mathbf{U}_k\mathbf{D}_k\mathbf{U}_k^{\mathrm{T}}$$

其中 $\mathbf{U}_k$ 是一个单位上三角矩阵，$\mathbf{U}_k^{\mathrm{T}}$ 是其转置，$\mathbf{D}_k$ 是个对角线矩阵。替代 RLS 中更新 $\mathbf{P}_k$ 的是更新其因子 $\mathbf{U}$ 和 $\mathbf{D}$。在指导手册的 CD（参见前言）中包含了一个 C 语言的 UD 算法程序。

10.6　应用举例 1——人脑电图中视觉伪像的自适应滤波

10.6.1　生理学问题

人的 EEG 描述了脑部的电活动，包含了诊断许多种神经错乱病症（neurological disorder）的有用信息。通常的 EEG 信号是从头皮上放置的电极上测量到的，一般具有很小的幅度，大约在 20 μV 左右。与许多生物医学信号一样，一些大的信号或人工运动会使 EEG 变得不可靠，降低其临床用途。例如，眨眼或眼球运动在眼睛周围产生大的电压，叫做眼电图（electrooculogram, EOG）。EOG

在头部的传播使EEG被污染，即所谓的视觉伪像（ocular artefact, OA）。测量的EOG和相应的被污染EEG信号的例子参见图10.16。

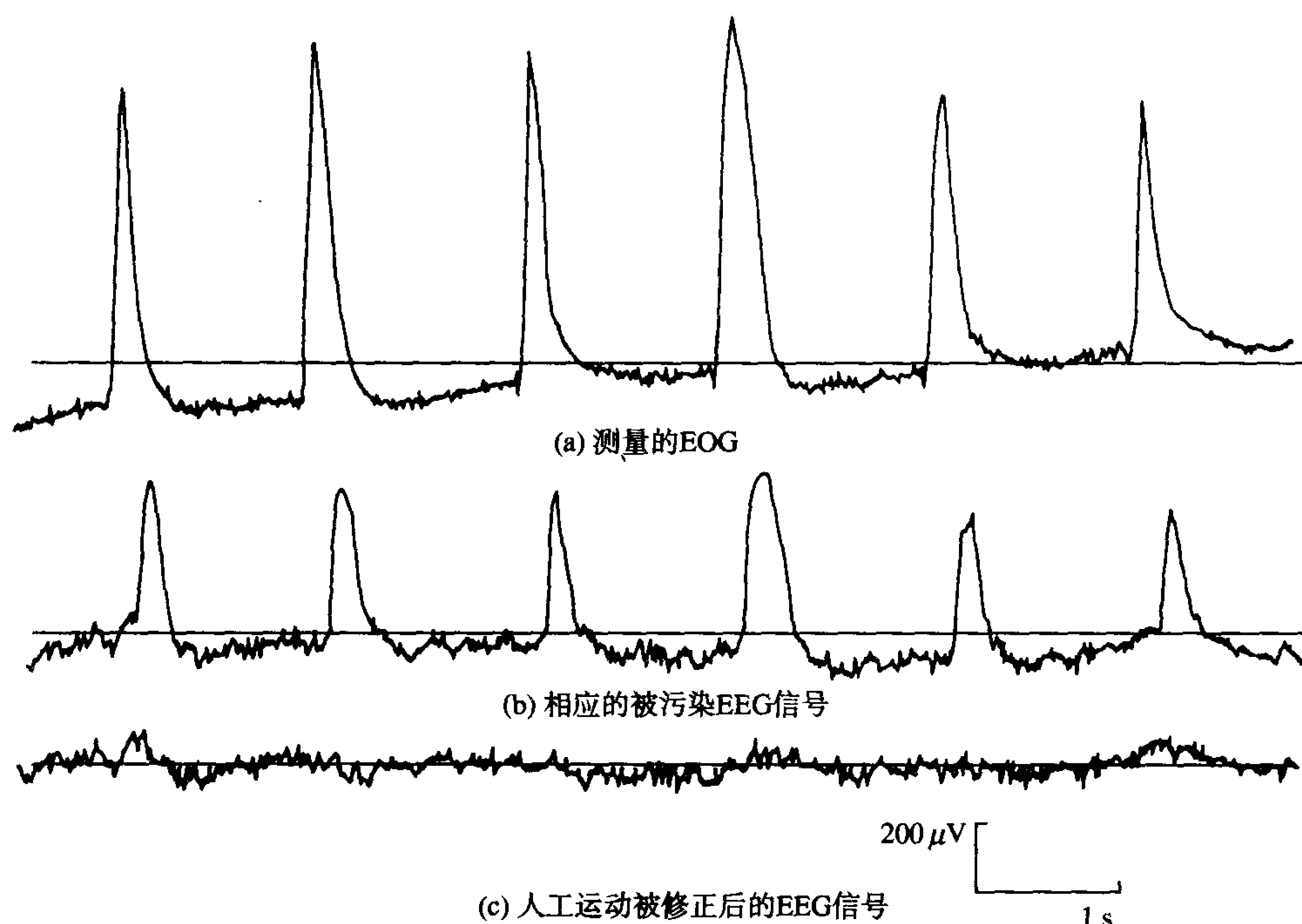


图10.16 脑电图中的视觉伪像问题

视觉伪像是区分正常与非正常脑活动的主要困难来源。在某些情况下，例如脑损伤婴儿和正面脑瘤病人，很难将EEG中相关的病理慢波与OA区分开。OA与临床感兴趣的信号间的相似性还使计算机自动分析EEG变得困难。一般来说，神经错乱病例通常在EEG中表现为慢波。遗憾的是，这种慢波不仅形状与OA很相似，且占据了相同的频带。问题归结为在消除OA的同时保留临床感兴趣的信号。

10.6.2 运动处理算法

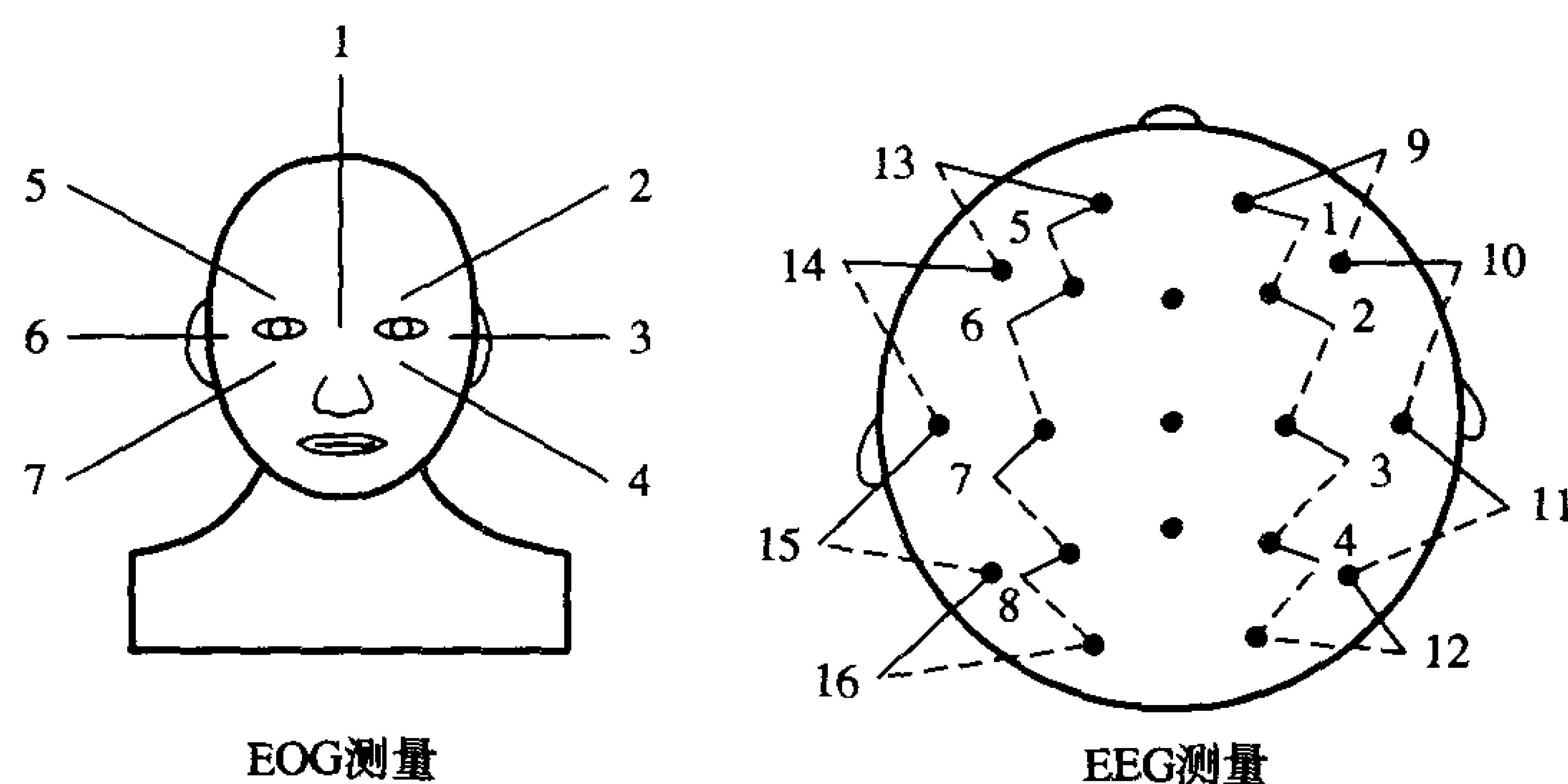
人们提出了几种方法用于处理OA。然而，考虑到如临床实验需求、实时应用限制、费用、OA的随机性质以及OA与某些脑溢血指示信号的频谱重叠等因素，OA处理应是自适应和实时的。

一个自适应视觉伪像滤波框架如图10.17所示。这种方法对OA的估计是从适当地测量EOG中获得的。然后从被污染的EEG中消去OA估计来产生无污染（视觉伪像）的EEG信号。为了阐明原理，考虑一个使用四个EOG信号源（参见图10.17(b)），从一个单独的EEG通道中消除视觉伪像污染的简单例子。被污染的EEG包含的信息y_k和EOG的四个测量值$x_k(0)$到$x_k(3)$参与估计视觉伪像$\sum_{i=0}^{3} w_k(i)x_k(i)$。再从被污染的EEG中消去OA估计得到无污染的EEG信号e_k：

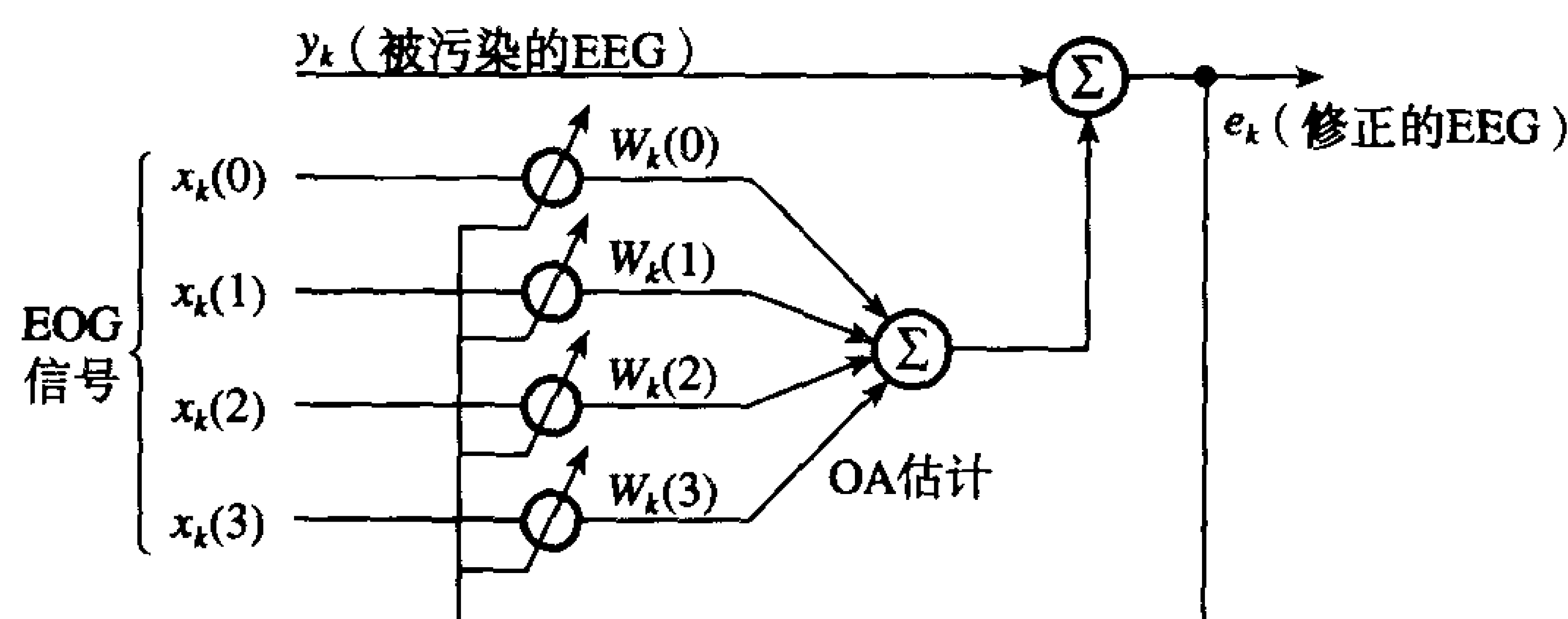
$$e_k = y_k - \sum_{i=0}^{n-1} w_k(i)x_k(i) \tag{10.37}$$

其中$w_k(i)$（$i = 0, 1,..., n-1$）是自适应滤波器的系数，也代表了EOG中传播到EEG的分量。e_k还用于调整自适应滤波器的系数（权重），并使用数值稳定的递归最小二乘算法，从而获得最优的OA估计。考虑到由于视觉伪像的变化使OA相应变化，连续地调整$w_k(i)$是必需的。

用于消除OA的自适应滤波算法可使用前面提到的UD算法。在LMS中常使用这种数值稳定的RLS算法，因为它具有较好的收敛时间，使其能根据不同种类的OA，产生所需的最优系数组来进行有效的消除。一个自适应校正EEG信号中人工运动的图例请参见图10.16(c)。



(a) 为测量视觉伪像EOG及EEG可能放置电极的位置



(b) 自适应视觉伪像滤波器

图10.17　自适应视觉伪像消除方法

10.6.3　实时实现

一个使用前面介绍的UD算法，在线消除视觉伪像的微处理器系统已经提出（Ifeachor et al., 1986）。该系统能实现一系列用户可选的模型。这个系统曾被应用于几个正常和患病的试验者，从患病的试验者那里获得了多个种类信息的良好结果。

但是，还发现当某些病理波（如慢波、癫痫病尖峰和复合型波）在EEG和EOG的电极上均被拾取，校正后的EEG波形幅度降低的现象。这是由于EOG被消除的程度取决于EOG和它在EEG中所含分量的相关度，且与OA形状相似的慢波的存在会导致被消除分量的大小不仅与EOG有关，还与慢波有关。因此，有必要使用一个专家系统来区分OA和慢波（Ifeachor et al., 1990）。

10.7　应用举例2——自适应电话回声对消

在早期通信系统中，当信号遇到阻抗不匹配时就会出现回声。图10.18(a)给出了一个简单的长距离电话电路。交换台的混合网络将用户的双线转换成四线电路，且为（信号的）双向传输提供不同的通道。这主要出于经济原因，如允许复用，即同时传输多个电话。

理想情况下，语音信号从用户A发出，沿着上面的传输路径到达右面的混合网络，再从那里传输到用户B，同时从B沿着下面的传输路径到达A。每端的混合网络需要保证从远方用户传来的语音信号耦合进它的双线端而不是到它的四线输出端。然而，由于阻抗失配，混合网络会让少部分

输入信号泄漏到输出路径，从而使讲话的人听到回声。当电话是经过了一个很长的距离（例如使用了地球静止卫星）时，回声可能延迟长达540 ms，从而给用户使用带来不良效果。传输距离越长，效果越差。为了克服这一问题，将回声对消器安装在网络两端，如图10.18(b)所示。

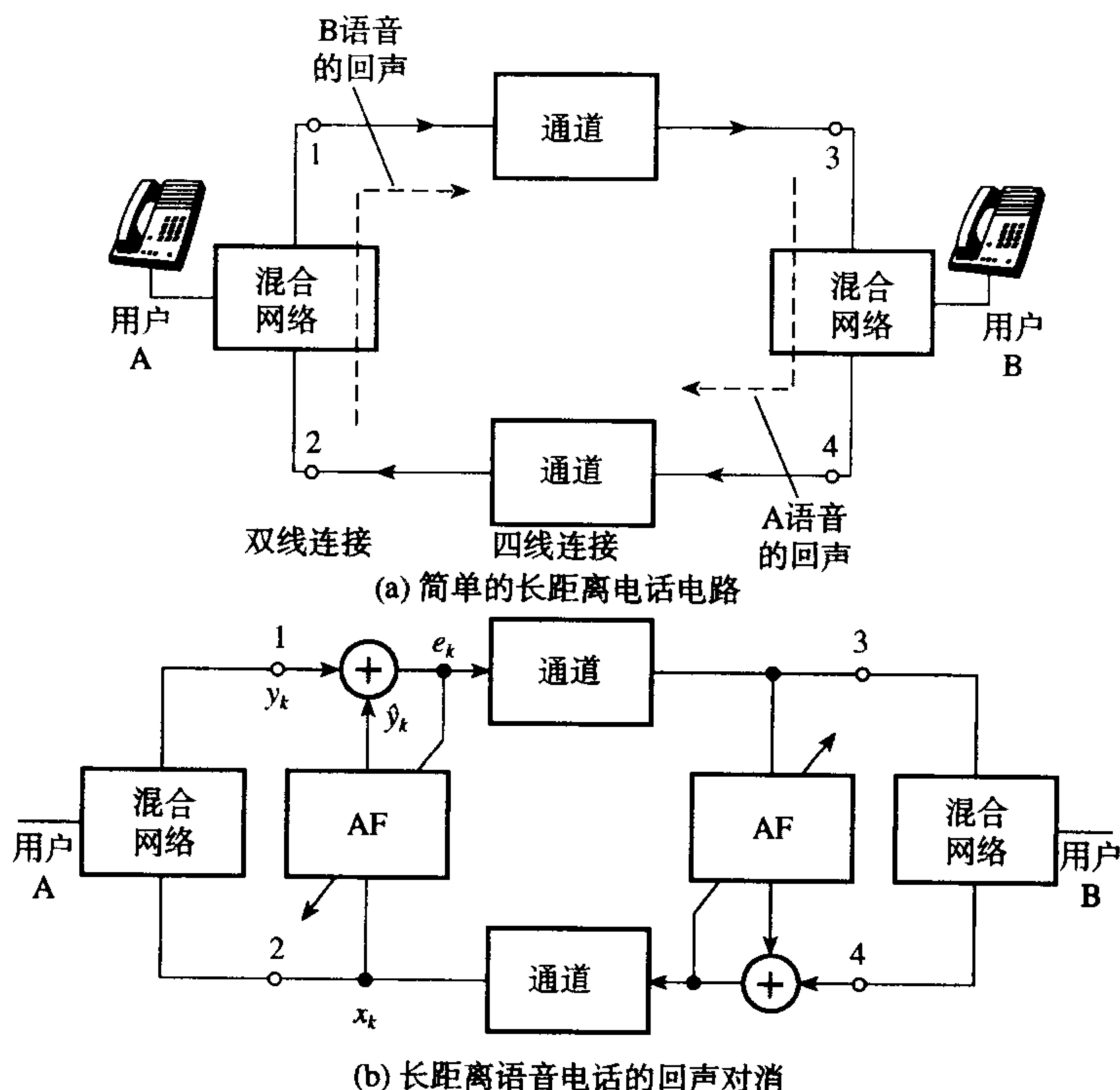


图10.18 简单的长距离电话电路和长距离语音电话回声对消

在通信系统的每一端（参见图10.18(b)），输入信号x_k被同时加在混合电路和自适应滤波器上（Duttweiller, 1978）。通过对回声进行估计并从返回信号y_k减去该估计以达到消除的目的。这里暗含的假设是回声路径（通过混合网络）为线性和时不变的。由此返回信号在k时刻可表示为

$$y_k = \sum_{i=0}^{N-1} w_k(i) x_{k-i} + s_k \tag{10.38}$$

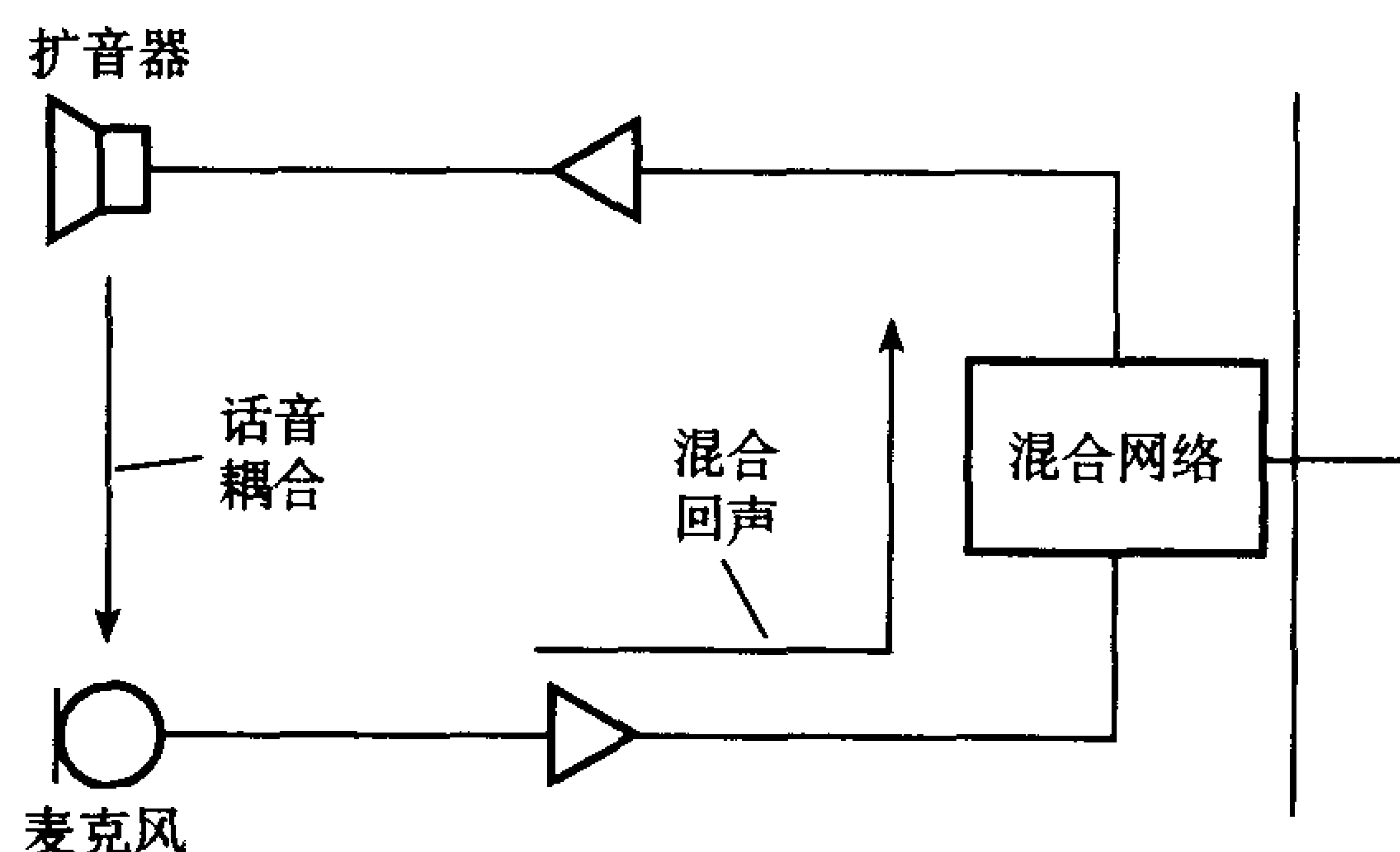
这里x_k是输入信号（从远端讲话人）的抽样，s_k是近端讲话人加白噪声的信号抽样，w_k是回声路径的冲激响应。回声对消器估计冲激响应，并得到回声的估计$\hat{y}_k = \sum w_k(i) x_{k-j}$，再从常规的返回信号$y_k$中减去它。出于经济的考虑，抽样率不能过高，因此滤波器系数和输入数据的字长效应限制着对消器的性能提高。基本限制则来源于自适应滤波器的误校准及回声路径的非线性。

10.8 其他应用

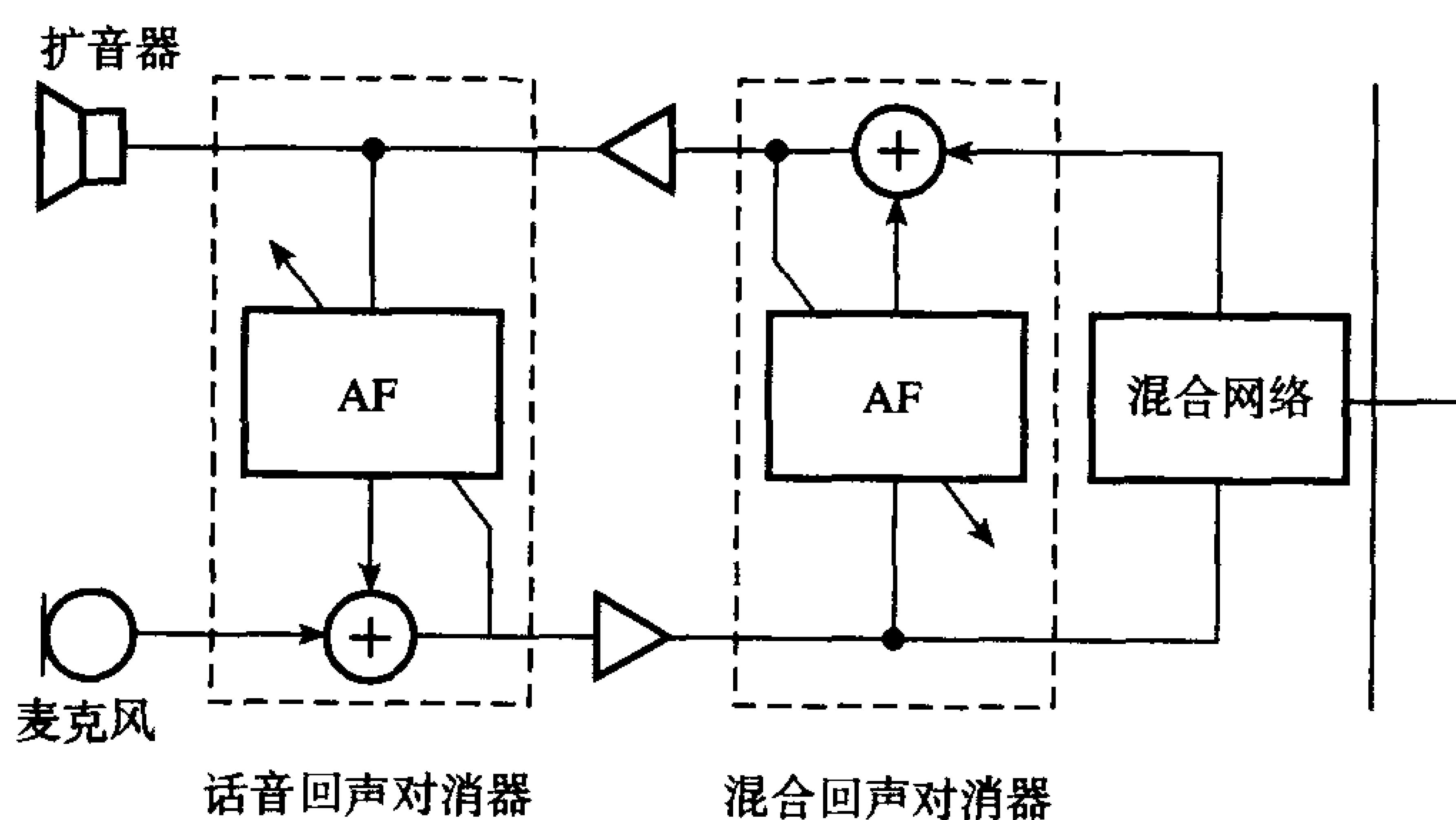
10.8.1 扩音电话

- 混合网络用于将传出和接收路径（从麦克风到扩音器）区分开，但在扩音器和麦克风之间存在着明显的话音耦合，这是二者距离较近和混合网络失配产生的泄漏所造成的（South et al., 1979）。

- 现在困难存在于如何为接收和传出方向提供适当的增益且不造成不稳定。
- 通常解决这一问题的方法是使用一个话音启动的转换开关来选择传出或接收路径，但这不能达到双工通信，因此令人难以满意。
- 一个较好的解决方案是采用自适应滤波技术来估计和控制话音和混合回声（参见图 10.19(b)）。这里的滤波器系数个数可以取得很大（如 512），因此需要使用快速的算法。
- 在远程会议网络（或公共播讲系统），话音反馈会产生与前面所述同样的问题。用于这些方面的自适应滤波器可能需要很大的滤波器系数个数（250 到 1000），尤其是在长回声反射时间的房间内，且必须快速收敛。



(a) 扩音电话



(b) 扩音电话中的话音和混合回声对消

图 10.19 扩音电话和其中的话音与混合回声对消

10.8.2 多径补偿

- 在一类扩频通信系统中，每个数据比特作为两个正交的 M 伪码序列比特之一被传输。传输哪个序列取决于数据比特是 0 还是 1。在接收端两个与发送端完全相同的序列与接收到的序列做互相关运算，由此判决接收到的数据比特是 1 还是 0。
- 当存在多径时，信号经过了不同的路径传输到接收机。这样的反射现象可能发生在山区或城市。接收到的信号包含了许多分量的和，它们的幅度和相位可以是不同的（参见图 10.20）。这降低了接收机的性能。
- 自适应滤波器用于估计总的多径响应和补偿多径效应。

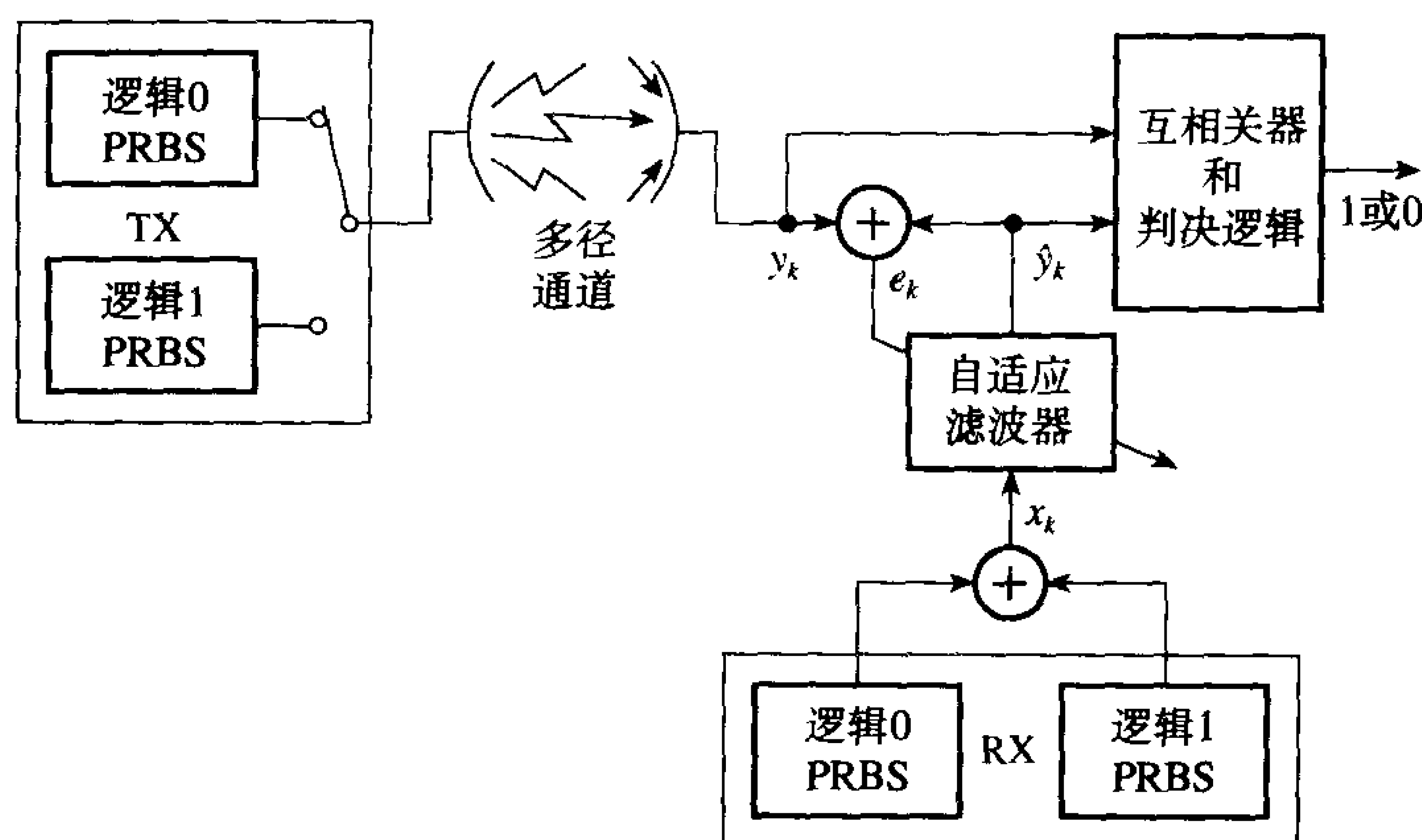


图 10.20　一个自适应扩频通信系统及多径效应补偿

10.8.3　自适应干扰源压制

- 在直接序列扩频中，经常要求抑制某个干扰源信号以提高接收机性能。自适应滤波器可以用于这个目的（参见图10.21）。在这种系统中，应用出发于干扰是高度相关的而伪码是弱相关的这一事实。由此滤波器的输出 y_k 是干扰的一个估计。从接收信号 x_k 中消去该估计，产生扩频信号的估计。
- 为了增强系统性能，使用一个两级干扰抑制器。自适应在线增强器，实际上是另一个自适应滤波器，抵消了导致希望信号被部分消除的有限相关效应。每个滤波器需要中等的系数个数（大约16），但抽样率必须大于400 kHz。

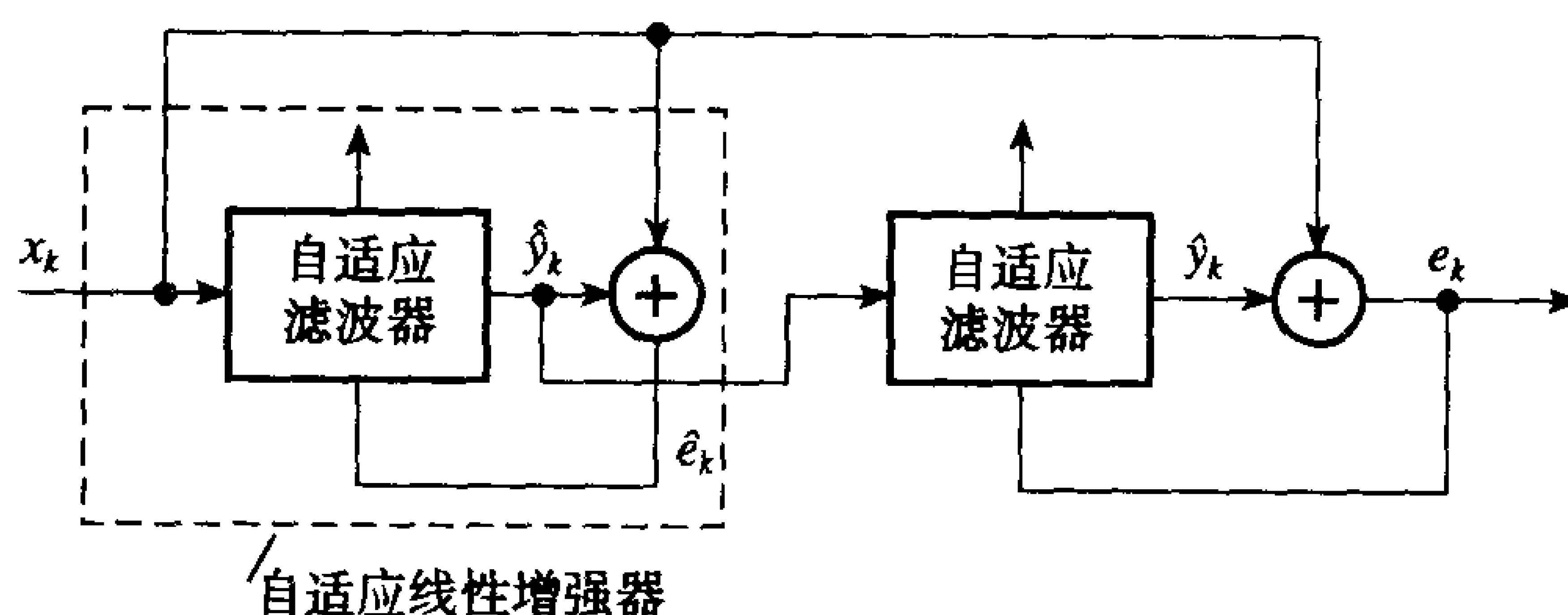


图 10.21　直接序列扩频接收机中的干扰源抑制

10.8.4　雷达信号处理

自适应信号处理技术被广泛应用于解决许多涉及雷达的问题。例如，自适应滤波器在单基地（有源）雷达中用于从希望目标信号中消除杂波分量。在地波雷达中，自适应滤波器用于降低HF频带的主要问题——通道互干扰。

10.8.5　从背景噪声中分离语音信号

在语音处理中，背景噪声是个严重的问题。自适应滤波器可以用于提高噪声环境（如作战飞机、坦克或车）中语音系统的性能，包括语言的清晰和可识别性。

10.8.6　胎儿监护——消除分娩期间母亲的ECG

- 从胎儿ECG中得到的信息，如胎儿心率模式，对评估孩子在出生前和出生过程中的状况很有价值。

- 从母亲的腹部上放置的电极获得的 ECG 由于被很大的背景噪声（如肌肉运动和胎儿运动）及母亲自己的 ECG 所污染而很不可靠。
- 自适应滤波器已被用于得到“无噪”的胎儿 ECG。图 10.22 给出了其实现方法。
- 四个胸导联用于检测胎儿的 ECG，一个或多个探头用来检测母亲和胎儿的混合 ECG。一个四通道自适应滤波器，每个通道 32 个系数，用于将母亲的心跳消除，如下图所示。

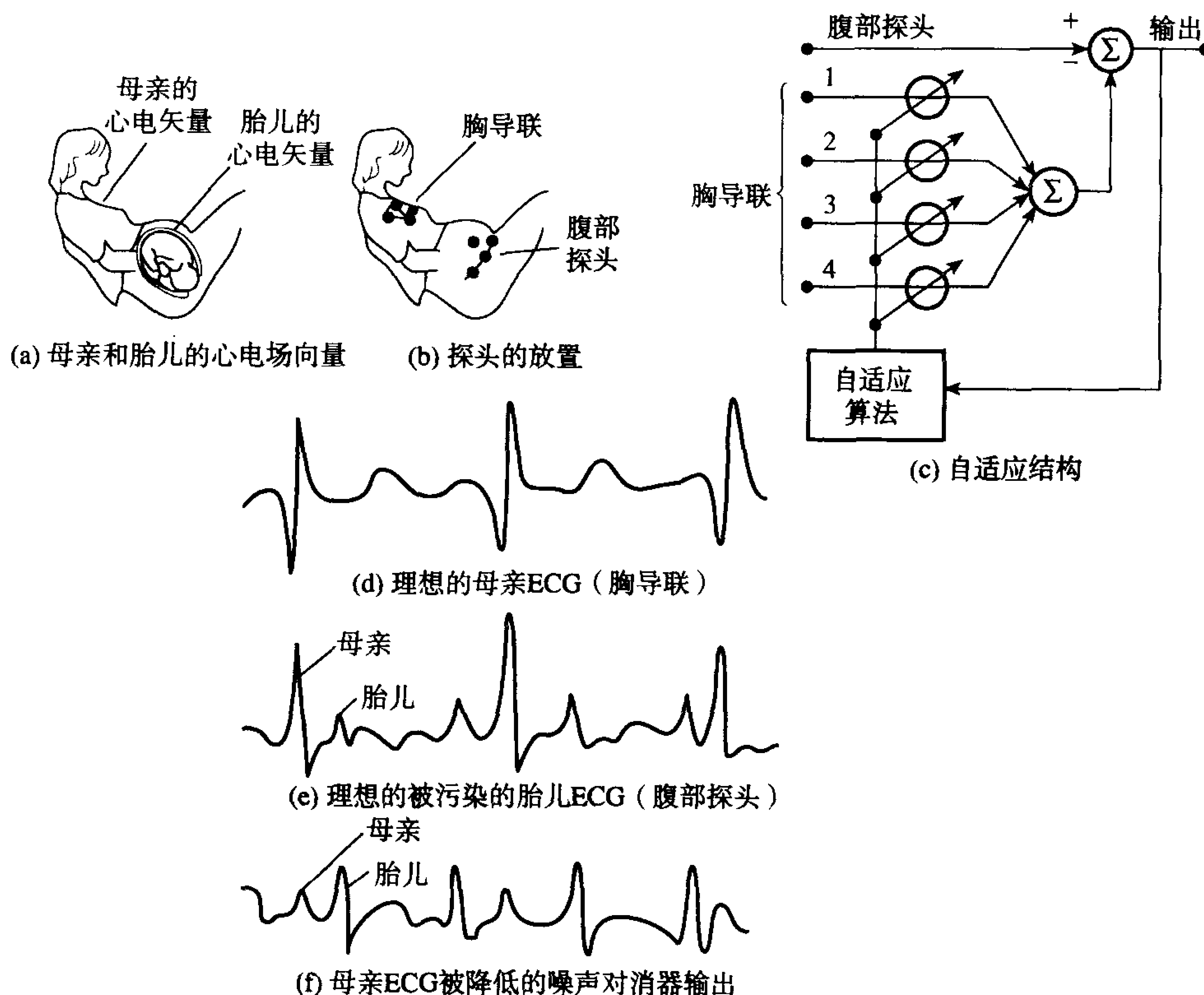


图 10.22　胎儿 ECG 中母亲 ECG 的自适应消除（Widrow et al., 1975a）

习题

10.1 在下面应用中，说明用自适应滤波器替代常规滤波器的理由：

(1) 从人的 EEG 中消除视觉伪像；

(2) 长距离电话的回声对消；

(3) 扩频通信中的干扰信号抑制。

从最速下降法出发：

$$\mathbf{W}_{k+1} = \mathbf{W}_k - \mu\nabla_k$$

其中，$\mathbf{W}_k$是离散时刻k滤波器的权重矢量，μ控制收敛速度和稳定性，∇_k是离散时刻k的误差性能曲面的真实梯度矢量。推导 Widrow-Hopf 的自适应噪声消除 LMS 算法，说明用到的任何合理假设。评论 LMS 算法在实际应用中的重要作用。

讨论LMS算法的两个主要使用限制和它们怎样降低了算法性能。建议怎样克服这些限制。

10.2 自适应噪声对消器的输出信号为

$$e_k = y_k - \mathbf{X}_k^{\mathrm{T}}\mathbf{W}_k$$

其中$\mathbf{W}_k$是自适应滤波器的权重矢量，其他变量具有通常的意义。从这个公式出发，推导

(1) 离散维纳－霍夫方程；

(2) 基本 LMS 算法。

说明任何用到的假设。

10.3 阐明当干扰信号x_k与被污染信号y_k不相关时，自适应滤波器会自动关闭。

10.4 通过框程图简要解释自适应噪声对消器的基本概念。探讨在你选择的某种实时应用中，自适应噪声消除的好处和不足，给出克服不足的方法。

参考文献

Bierman G.J. (1976) Measurement updating using the UD factorization. *Automatica*, **12**, 375–82.

Duttweiller D.L. (1978) A twelve-channel digital echo canceler. *IEEE Trans. Communications*, **26**, 647–53.

Ferrara E.R. and Widrow B. (1981) The time-sequenced adaptive filter. *IEEE Trans. Acoustics, Speech and Signal Processing*, **29**(3), 766–70.

Haykin S. (1986) *Adaptive Filter Theory*. Englewood Cliffs NJ: Prentice-Hall.

Ifeachor E.C., Jervis B.W., Morris E.L., Allen E.M. and Hudson N.R. (1986) A new microcomputer-based on-line ocular artefact removal (OAR) system. *IEE Proc.*, **133**, 291–300.

Ifeachor E.C., Hellyar M.T., Mapps D.J. and Allen E.M. (1990) Knowledge based enhancement of EEG signals. *Proc. IEE (Part F)*, **137**(5), 302–10.

Mansour D. and Gray A.H. (1982) Unconstrained frequency domain adaptive filter. *IEEE Trans. Acoustics, Speech and Signal Processing*, **30**, 726–34.

Peterka P. (1975) A square root filter for real-time multivariate regression. *Kybernetika*, **11**, 53–67.

South C.R., Hoppitt C.E. and Lewis A.V. (1979) Adaptive filters to improve loudspeaker telephone. *Electronics Lett*, **15**, 673–4.

Widrow B. and Winter R. (1988) Neural nets for adaptive filtering and adaptive pattern recognition. *IEEE Computer*, 25–30.

Widrow B., Glover J.R., McCool J.M., Kaunitz J., Williams C.S., Hearn R.H., Zeidler J.R., Dong E. and Goodlin R.C. (1975a) Adaptive noise cancelling: principles and applications. *Proc. IEEE*, **63**, 1692–716.

Widrow B., McCool J.M. and Ball M. (1975b) The complex LMS algorithm. *Proc. IEEE*, 719–20.

参考书目

Clark G.A., Mitra S.K. and Parker S.R. (1981) Block implementation of adaptive digital filters. *IEEE Trans. Acoustics, Speech and Signal Processing*, **29**, 744–52.

Clark G.A., Parker S.R. and Mitra S.K. (1983) A unified approach to time- and frequency-domain realization of FIR adaptive digital filters. *IEEE Trans. Acoustics, Speech and Signal Processing*, **31**, 1073–83.

Cowan C.F.N. and Grant P.M. (eds) (1985) *Adaptive Filters*. Englewood Cliffs NJ: Prentice-Hall.

De Courville M. and Duhamel P. (1995) Adaptive filtering in subbands using a weighted criterion. In *Proc. ICASSP*, Detroit, MI, Vol. 2, pp. 985–8.

Dentino M., McCool J.M. and Widrow B. (1978) Adaptive filtering in the frequency domain. *Proc. IEEE*, **66**, 1658–9.

Dudek M.T. and Robinson J.M. (1981) A new adaptive circuit for spectrally efficient digital microwave-radio-relay systems. *Electronics and Power*, 397–401.

Falconer D.D. (1982) Adaptive reference echo cancellation. *IEEE Trans. Communications*, **30**, 2083–94.

Ferrara E.R. and Widrow B. (1981) Multichannel adaptive filtering for signal enhancement. *IEEE Trans. Acoustics, Speech and Signal Processing*, **29**, 766–70.

Gilloire A. and Vetterli M. (1992) Adaptive filtering in subbands with critical sampling: analysis, experiments, and applications to acoustic echo cancellation. *IEEE Trans. Circuits and Systems*, **40**, 1862–75.

Harrison W.A., Lim J.S. and Singer E. (1986) A new application of adaptive noise cancellation. *IEEE Trans. Acoustics, Speech and Signal Processing*, **34**, 21–7.

Holte N. and Stueflotten S. (1981) A new digital echo canceler for two-wire subscriber lines. *IEEE Trans. Communications*, **29**, 1573–81.

Lappage R., Clarke J., Palma G.W.R. and Huizing A.G. (1987) The Byson research radar. In *International Conf. Radar 87*, October 1987, London: IEE, 453–61.

Levin M.D. and Cowan C.F.N. (1994) The performance of eight recursive least squares adaptive filtering algorithms in a limited precision environment. In *Proc. European Signal Processing Conf.*, Edinburgh, pp. 1261–4.

Lewis A. (1992) Adaptive filtering applications in telephony. *BT Technology J.*, **10**, 49–63.
Li Y. and Ding Z. (1995) Convergence analysis of finite length blind adaptive equalizers. *IEEE Trans. Signal Processing*, **43**, 2120–9.
Macchi O. (1995) *Adaptive Processing: The LMS Approach with Applications in Transmission*. New York: Wiley.
Messerschmitt D.G. (1984) Echo cancellation in speech and data transmission. *IEEE J. Selected Areas in Communications*, **2**, 283–97.
Mikhael W.B. and Wu F.H. (1987) Fast algorithms for block FIR adaptive digital filtering. *IEEE Trans. Circuits and Systems*, **34**, 1152–60.
Mueller K.H. (1976) A new digital echo canceler for two-wire full-duplex data transmission. *IEEE Trans. Communications*, **24**, 956–62.
Ochia K., Araseki T. and Ogihara T. (1977) *IEEE Trans. Communications*. **25**, 589–94.
Ogue J.C., Saito T. and Hoshiko Y. (1983) A fast convergence frequency domain adaptive filter. *IEEE Trans. Acoustics, Speech and Signal Processing*, **31**, 1312–14.
Reed F.A., Feintuch P.L. and Bershad N.J. (1985) The application of the frequency domain LMS adaptive filter to split array bearing estimation with a sinusoidal signal. *IEEE Trans. Acoustics, Speech and Signal Processing*, **33**, 61–9.
Saulnier G.J., Das P.K. and Milstein L. (1985) An adaptive digital suppression filter for direct sequence spread spectrum communications. *IEEE J. Selected Areas in Communications*, **3**, 676–86.
Sethares W.A., Lawrence D.A., Johnson C.R. and Bitmead R.R. (1986) Parameter drift in LMS adaptive filters. *IEEE Trans. Acoustics, Speech and Signal Processing*, **34**, 868–79.
Sondhi M.M. and Berkley D.A. (1980) Silencing echoes on the telephone network. *Proc. IEEE*, **68**, 948–63.
Tao Y.G., Kolwicz K.D., Gritton C.W.K. and Duttweiller D.L. (1986) A cascadable VLSI echo canceller. *IEEE Trans. Acoustics, Speech and Signal Processing*, **34**, 297–303.
Thornton C.L. and Bierman G.J. (1978) Filtering and error analysis via the UDU covariance factorization. *IEEE Trans. Automatic Control*, **23**, 901–7.
Widrow B. (1966) *Adaptive Filters I: Fundamentals*. Report SU-SEL-66-126, Stanford Electronics Laboratory, Stanford University, CA.
Widrow B. (1971) Adaptive filters. In *Aspects of Network and System Theory* (Kalman R. and DeClaris N. (eds)), pp. 563–87. New York: Holt, Rinehart and Winston.
Widrow B. (1976) Stationary and nonstationary learning characteristics of the LMS adaptive filter. *Proc. IEEE*, **64**, 1151–62.
Widrow B. and Stearns S.D. (1985) *Adaptive Signal Processing*. Englewood Cliffs NJ: Prentice-Hall.
Widrow B., Mantey P., Griffiths L. and Goode B. (1967) Adaptive antenna systems. *Proc. IEEE*, **55**, 2143–59.

附录

10A 自适应滤波的C语言程序

下面是几个C语言编程的自适应算法，能够用在本章所述的内容上：

(1) lmsflt.c，LMS算法。

(2) uduflt.c，UD算法。

(3) sqrflt.c，平方根算法和

(4) rlsflt.c，递归最小二乘算法。

版面所限，只列出第一个源程序（参见程序10A.1）。不过，本书的配套指导手册*A Practical Guide for MATLAB and C Language Implementations of DSP Algorithms*（详见前言）提供的CD上包含所有的程序。

程序10A.1 LMS算法的C语言实现程序（lmsflt.c）

```
/*-----------------------------------------------------------------------------------------*/
/*      implementation of the LMS algorithm                                                */
/*                                                                                         */
/*      manny 6.11.92                                                                      */
/*                                                                                         */
/*      inputs:                                                                            */
/*      x[]         input data vector                                                      */
/*      dk          latest input data value                                                */
/*      w[]         coefficient vector                                                     */
/*                                                                                         */
/*      outputs:                                                                           */
/*      ek          error value                                                            */
/*      yk          digital filter output                                                  */
/*      w[]         updated coefficient vector                                             */
/*-----------------------------------------------------------------------------------------*/
double    lmsflt()
{
          int       i;
          double    uek,yk;

          yk = 0;
          for(i=0; i<N; ++i){                   /* digital filtering */
                    yk=yk+w[i]*x[i];
          }
          ek=dk-yk;                             /* compute output error */

          uek=2*mu*ek;                          /* update the weights */
          for(i=0; i<N; i++){
                    w[i]=w[i]+uek*x[i];
          }
          return(yk);
}
```

为了说明怎样实现自适应滤波器，我们使用上面列出的程序10A.1来在宽带噪声中检测一个音调。

10A.1　噪声中窄带信号的自适应增强

自适应滤波器常用于检测或增强宽带噪声淹没下的窄带信号。其通常的结构如图 10A.1 所示。它包括一个延迟单元（用 z^{-M} 表示）和一个自适应预测器。延迟单元用于消除噪声分量间可能存在的任何相关。自适应预测器实际上是一个具有可调系数的 FIR 滤波器，其输出 y_k 为增强后的窄带信号。在某些应用中，自适应滤波器的第二输出 e_k（而不是 y_k）是所希望的输出。预测系数 $w_k(i)$，用一个适合的自适应算法来优化，在我们的例子中使用 LMS 算法（详见 10.4 节）。

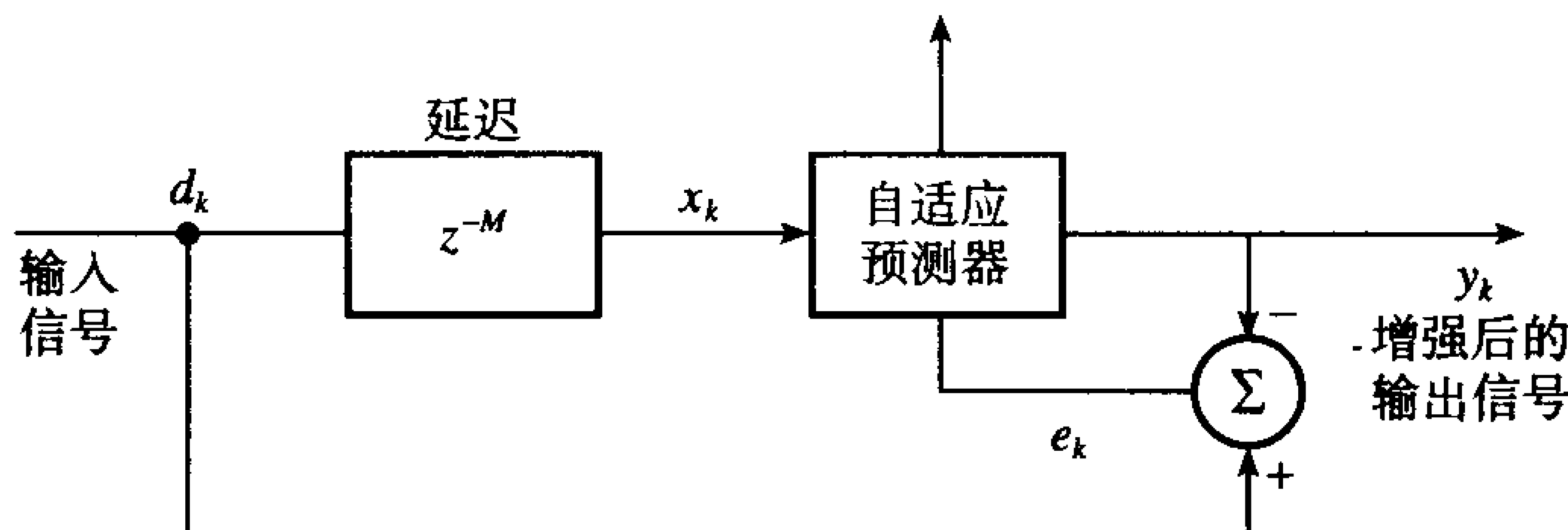


图 10A.1　自适应信号增强

如果使用 LMS 算法，自适应滤波器可用下面的方程来描述：

$$y_k = \sum_{i=0}^{N-1} w_k(i)x_k(i) \tag{10A.1}$$

$$e_k = d_k - y_k \tag{10A.2}$$

$$w_{k+1}(i+1) = w_k(i) + 2\mu e_k x_k(i), \quad i = 0, 1, \ldots, N-1 \tag{10A.3}$$

其中 d_k 是噪声中的窄带信号抽样，$x_k(i)$ 是由 d_k 延迟所获得的输入数据矢量，$w_k(i)$ 是第 k 抽样时刻的预测系数矢量，μ 是稳定因子，y_k 是增强后的窄带信号。

程序 10A.1 中的函数 lmsflt.c 是上面方程的一个 C 语言实现程序。程序 10A.2 中列出的是如何使用 lmsflt.c 来做信号增强。为了仿真该问题，一个宽带噪声加在 500 Hz 的正弦波信号上，混合数据存入 ASCII 格式的文件 din.dat 中。再将含噪的正弦波送给自适应滤波器。为了仿真实时自适应滤波，输入数据是按每一时刻一个抽样从文件中读出和送给自适应滤波器的。对于很大量的数据，使用者可以按块读取数据以提高效率。图 10A.2 给出了基于 LMS 滤波器的结果。

从图 10A.2 明显看出，使用自适应算法，使用者必需确定自适应滤波器的参数，例如 FIR 滤波器系数个数 N、延迟因子 M 和稳定因子 μ，还要注意输入数据的格式。例如，在某些应用中，输入数据是从多通道源获得的，每个 $x_k(i)$ 单元代表着从一个通道得到的数据。在这样的情况下，自适应算法的输入数据阵列需要经过适当的修正。

程序 10A.2　自适应信号增强程序

```
/* ------------------------------------------------------------------------------------ */
/*                                                                                      */
/*      program to illustrate adaptive filtering using                                  */
/*      the LMS algorithms                                                              */
/*                                                                                      */
/*      program name: adfilter.c                                                        */
/*                                                                                      */
/*      manny, 7.11.92                                                                  */
/* ------------------------------------------------------------------------------------ */
```

```
#include        <stdio.h>
#include        <math.h>
#include        <dos.h>

/* constant definitions */

#define N       30      /* filter length */
#define M       1       /* delay */
#define w0      0       /* initial value for adaptive filter coefficients */
#define npt     N+M
#define SF      2048    /* factor for reducing the data samples – 11 bit ADC assumed */
#define mu      0.04

double      lmsflt();
void        initlms();
void        update_data_buffers();
void        initfiles();
float       x[npt], d[npt], dk, ek;
double      w[npt];
FILE        *in,*out,*fopen();
char        din[30];

main()
{
            double yk, yk1;

            initfiles();
                                                    /* lms-based adaptive filter */
            initlms();
            while(fscanf(in,"%f",&dk)!=EOF){
                    dk=dk/SF;
                    update_data_buffers();
                    yk=lmsflt();
                    yk1 =SF*yk;
                    fprintf(out,"%lf \n",yk1);
            }
            fcloseall();
}

/* ------------------------------------------------------------------------------------------------------ */
void        initfiles()
{
            clrscr();
            printf("enter name of file holding data to be filtered \n");
            scanf("%s",din);
            printf("\n");
            printf("the filtered data will be stored in dout.dat \n");

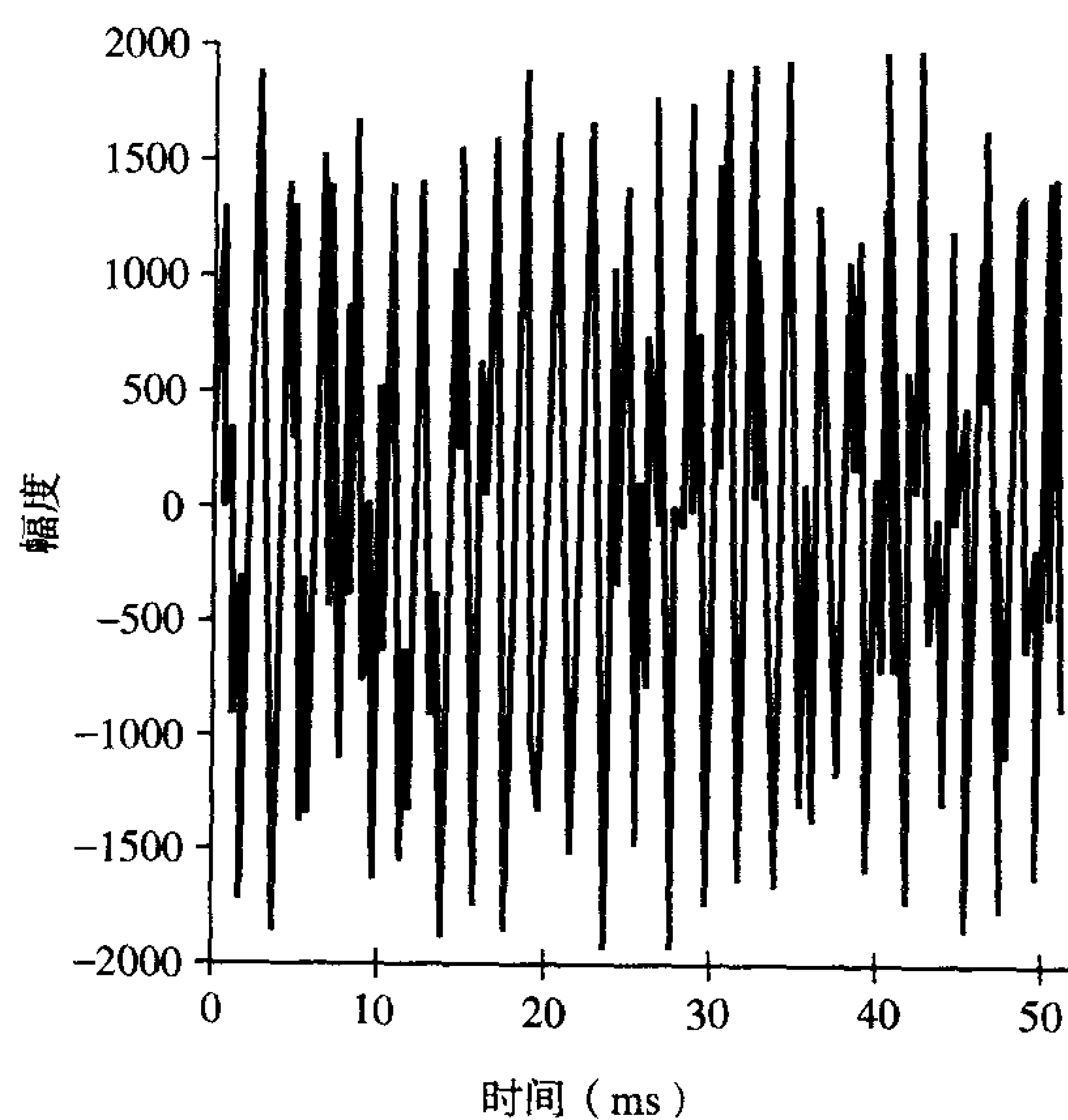
            if((in=fopen(din,"r"))= =NULL){
                    printf("cannot open input data file \n");
                    exit(1);
            }
            if ((out = fopen ("dout.dat", "w"))= =NULL){
                    printf("cannot open output data file \n");
                    exit(1);
            }
            return;
}
```

```
/*---------------------------------------------------------------------------------------*/

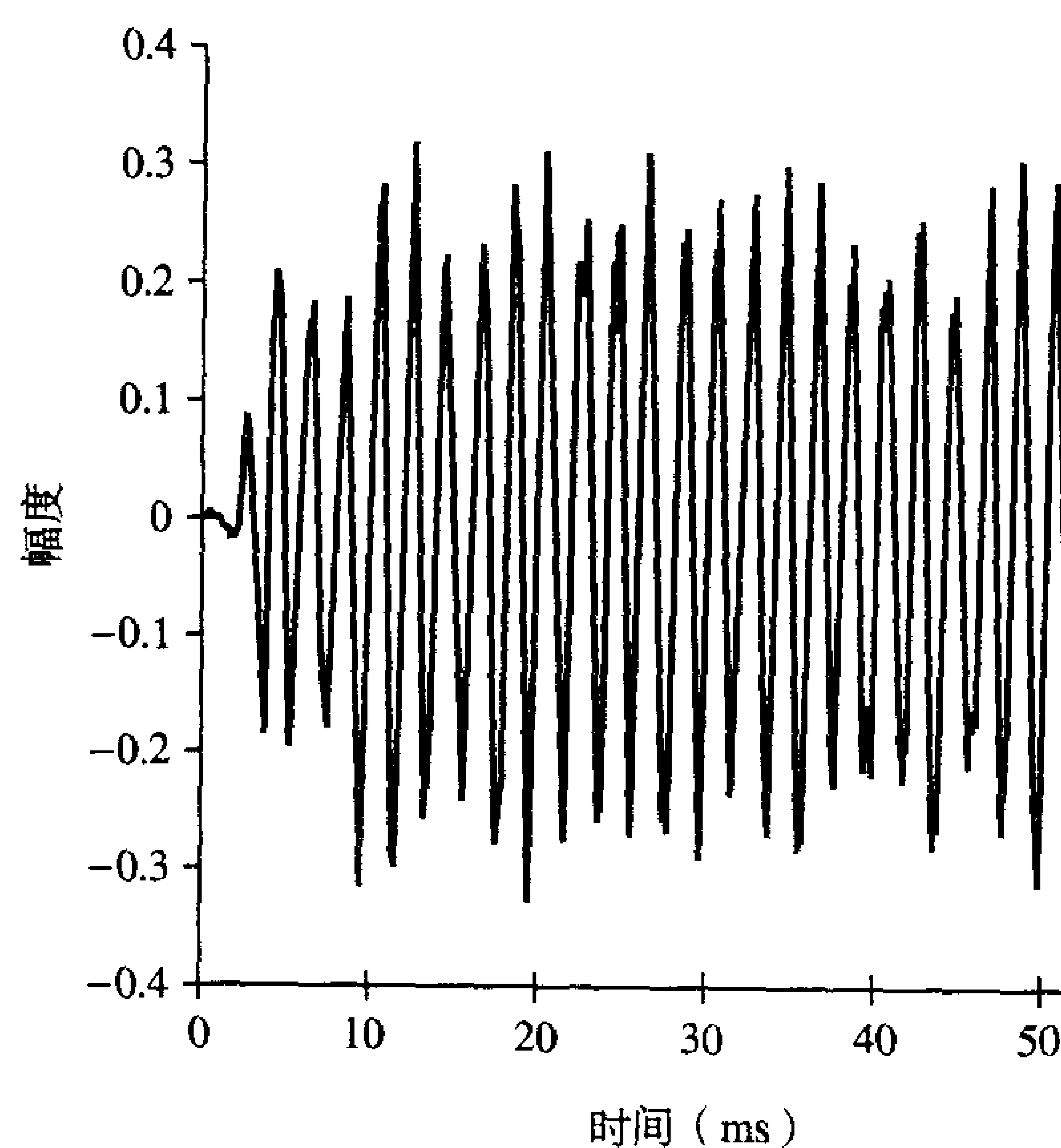
void        update__data__buffers()
{
            long     j, k;
            for(j=1; j<N; + +j){                          /*update x-data buffer*/
                  k=N-j;
                  x[k]=x[k-1];
            }
            x[0]=dk;
            if(M>0)
                  x[0]=d[M-1];
            for(j=1; j<M; + +j){                          /*update d-data buffer*/
                  k=M-j;
                  d[k]=d[k-1];
            }
            d[0]=dk;
}

/*---------------------------------------------------------------------------------------*/
void        initlms()
{
            long     i;
            for(i=0; i<npt; + +i){
                  x[i]=0;
                  d[i]=0;
                  w[i] = w0;
            }
}
/*---------------------------------------------------------------------------------------*/

#include          "lmsflt.c";
```



(a) 含噪信号



(b) 增强后的信号

图 10A.2　一个窄带信号的自适应增强

10B 自适应滤波的 MATLAB 程序

MATLAB并不直接支持自适应滤波。然而，我们用MATLAB开发了两个基本的自适应算法——LMS 和 RLS 算法：

lmsadf.m ——执行基于 LMS 自适应滤波的函数
rlsadf.m ——执行基于 RLS 自适应滤波的函数

两个程序的演示例子可在指导手册（详见前言）中找到。

第11章　频谱估计与分析

本章包含了下面的课题：频谱分析的基本概念；需要小心的陷阱，尤其是在非参数法中；数据窗的性质；数据的预处理和数据窗函数的选择。并且描述了非参数法，比较了修正周期图法、基于数据自相关函数的快速傅里叶变换的布莱克曼－图基（Blackman-Tukey）法和快速相关法。同时给出了最常用的基于自回归模型的参数频谱估计方法的描述，以及其他参数法的简介等。本章的应用介绍包括对脑电波的处理以区分脑部患病和普通的试验者，以及基于自回归模型的人脑电图分析。

11.1　引言

在3.1节介绍了怎样将数据从时域转换到频域。在频域进行频谱分析与估计的原理和实践是本章的目标。谐波幅度或相位对频率变化的图经常产生一种对数据或波形的更为复杂的表示，特别是当后者具有随机性质时。通过按照某些适合的准则来选择特定的谐波而拒绝其他，数据量可能进行显著的压缩。频谱分析有多种应用，如通信工程信号、脑病诊断中的事件关联或驱动的人脑电图（Jervis et al., 1993）及其他生物医学信号、气象数据的研究，还有工业过程控制和最优线性滤波器设计中的噪声谱测量等。

频谱估计技术可分为两大类——非参数法和参数法。非参数法包括周期图、巴特利－韦尔奇（Bartlett-Welch）的修正周期图，以及布莱克曼－图基法。所有这些方法具有可以利用快速傅里叶变换的优势，但缺点是当数据长度较短时，频率分辨率较低。另外，还需特别注意以获得有意义的结果。参数法则可以提供高的分辨率，且计算效率较高。但是需要构建一个足够精确的过程模型以估计频谱。最常用的参数法是从一个信号的自回归模型的参数来推导频谱。自回归频谱估计的介绍请参见11.5节。

在非参数频谱分析中需要避免许多陷阱，相关的问题包括混叠、扇形损失、有限字长、频谱泄漏和频谱混淆，在11.3节将予以介绍和讨论。

频谱泄漏和频谱混淆的不良作用可以通过对数据加一个适合的函数窗来降到最低。抽样数据值与所选择的窗函数的抽样值点依次相乘。加窗课题将在11.3.2节讨论。等效噪声带宽、处理增益、最坏处理损失和最小分辨带宽是选择某个合适的窗函数时应该考虑的性质（参见11.3.2.1节）。在讲述重叠相关的章节中，将会看到把数据分隔成许多部分的加窗并计算它们的谱平均，这比直接计算加窗数据的频谱能明显提高对频谱的估计。11.3.2.1节的数据窗偏差效应部分给出了信号能量是怎样损失的，以及通过对数据加窗前的预处理来克服数据窗的偏差效应。

频谱估计的质量评估是基于估计理论，因此将阐述该理论的部分基本概念。统计估计包括根据总体的样本来确定统计量的期望值。但是，在时间序列分析中，作为时间的函数获得的离散数据比从总体中同时获得抽样更为常见。这个困难通常采用假定该随机过程是遍历的来避免，即从时间序列获得的数据和假设的抽样具有完全一致的性质。一些统计学定义现在依次给出。

一个时间序列的数据值$x(n)$（$n = 0, 1, \ldots, N-1$），其均值为$x(n)$的期望值$E[x(n)]$，即

$$E[x(n)] = \frac{1}{N}\sum_{n=0}^{N-1} x(n) \tag{11.1}$$

其中E代表取数学期望。同样时间序列的方差为

$$\mathrm{var}[x(n)] = E\{[x(n) - \bar{x}(n)]^2\} \tag{11.2}$$

$x(n)$的协方差为

$$c_{xx}(m) = E\{[x(n) - \bar{x}(n)][x(n+m) - \bar{x}(n)]\} \tag{11.3}$$

这里m代表数据点的延迟而$\bar{x}(n)$代表$E[x(n)]$。从有限实现估计到的功率谱为

$$P_{\mathrm{E}}(\omega) = \sum_{m=-\infty}^{\infty} c_{xx}(m)\exp(-\mathrm{j}\omega m) \tag{11.4}$$

注意，如果分析的是有限长的波形而不是无限长的随机过程，那么使用能量谱密度将更为合适。功率谱密度的单位是$\mathrm{V^2\,Hz^{-1}}$。如果统计量的估计为α，则估计的偏差定义为真实（总体）值与估计值的差：

$$\mathrm{bias} = \alpha - E[\alpha] \tag{11.5}$$

如果偏差为零，则估计得到了真实值；如果偏差不为零，它代表了α的误差，称α的估计是有偏的。显然好的估计应是无偏的。α的方差是α的概率密度分布函数的尖峰宽度的一个测度。小的方差对应着窄的尖峰，当方差趋近于零且估计是无偏的时，估计值将接近于总体（真）值。如果随着数据长度N的增长，方差趋近于零，则称估计是一致的。如果估计不是一致的，则在不同的实现（样本）之间，随着数据长度的增长，估计的波动越来越大。因此，通常希望估计既是无偏的又是一致的。

频谱估计的周期图法的性质在11.3.3节和11.3.4节讨论。可以发现从周期图得到的频谱估计是不一致的，即连续的实现产生上下波动的估计，且只有相当多的数据才能得到无偏估计。稳定和更为精确的估计来自于对数据分隔加窗再将谱平均。相关的方法有著名的巴特利-韦尔奇的修正周期图法。最后介绍的非参数法是布莱克曼-图基法。先计算数据加窗后的自相关函数，再从它的FFT变换得到能量谱。布莱克曼-图基频谱估计的特点是比其他周期图法大得多的品质因数。

在进行功率谱密度估计时，希望确定品质因数以允许比较不同的估计。功率谱密度的均方与方差之比通常将其建议为适合的品质因数（Proakis and Manolakis, 1989）：

$$Q = \frac{\{E[P_{\mathrm{E}}(f)]\}^2}{\mathrm{var}[P_{\mathrm{E}}(f)]} \tag{11.6}$$

11.2　频谱估计原理

在本节首先考虑一个时域上的电压波形。波形的形状可以提供有用的信息。例如，它可以是一个正弦波，用幅度、频率和相位角就可以准确地刻画。为了更明确，可以将其描述为波形在已知的频率上包含具有特定幅度和相位的单一分量。或者说，使用两个图形：一个是幅度-频率，另一个是相位-频率，来表示电压-时间的波形。因为正弦波只有一个幅度、一个相位，所以一个频率幅度或相位图形上只有一个点。从傅里叶分析（参见第3章）中可知，所有波形从数学上都可以表示为大量正弦波的合成，每个正弦波都是在特定频率上具有特定幅度和相位。这样，所有时域波形都可以用一个幅度频率图与一个相位频率图来表示。这些图形称为幅度谱和相位谱。这些谱非常重要，因为它们提供了表示波形的一种补充方式，并且可以更清楚地揭示波形所包含的频率内容。对于谱形状的观察和它们的变化经常有助于对波形的理解和解释。幅度和相位谱通常比波形能提供更多的有用信息。关于从时域转换到频域和反向转换的课题在第3章已做过介绍。使用傅里叶序列和复傅里叶序列将周期波形转换到频域也已经回顾了。从中可以看出，周期波形的正弦频率分量的频率，即所谓的傅里叶分量是互为谐波关系的，换句话说，每个都是第一个谐波分量f的整数倍，其中

$$f = 1/T_p$$

这里T_p是波形的重复周期。它进一步说明谐波分量是按频率间隔$f = 1/T_p$均匀分布的，分隔在这个意义上称为频率分辨率。幅度谱的幅度测量单位是伏特。作为例子，图3.1(a)给出了一个周期电压脉冲波形，而图3.1(b)和图3.1(c)分别显示这个波形的幅度谱和相位谱。第3章的引言提到了幅度和相位谱的各种不同用途。

非周期性但是连续的信号，可以使用3.1.2节介绍的傅里叶变换从时域转换到频域。变换后“幅度”具有的度量为V Hz^{-1}，因此当对频率轴做图时称为幅度谱密度。所以两个频率间隔的曲线下的面积产生了波形在两个频率间频率分量的“平均”电压。将傅里叶变换后的“幅度”平方得到电压波形的能量谱密度，单位为J Hz^{-1}。术语“谱”一般是指能量谱密度对频率的图形。图3.2(a)和图3.2(b)分别显示了一个长方形脉冲的幅度和能量谱密度。在第3章还介绍了怎样使用离散傅里叶变换（DFT）计算谱的抽样与非周期性电压波形。看到DFT分量是以谐波关系依赖于第一个谐波角频率$\Omega = 2\pi/(N-1)T$，所以第一个谐波频率f是

$$f = 1/(N-1)T \tag{11.7a}$$

这里N是数据的个数，T是抽样间隔。由于$(N-1)T = T_p$，即抽样波形的时间长度，第一个谐波频率也可表示为

$$f = 1/T_p \tag{11.7b}$$

同样，由于互为谐波关系，谱的频率分辨率也是$1/T_p$。因此波形越长，谱的频率分辨率越高。

3.2节给出了一个例子，计算数据序列{1, 0, 0, 1}的DFT。数据序列的DFT结果是{2, 1+j, 0, 1−j}。因此，第二个谐波分量1+j的幅度为$\sqrt{(1^2+1^2)} = \sqrt{2}$。如果数据序列代表了电压的抽样，则第二个谐波的幅度为$\sqrt{2}$ V，能量为$(\sqrt{2}\ \mathrm{V})^2$，或者2焦耳。相应的相位角为$\tan^{-1}$（虚部分量/实部分量）$= \tan^{-1} 1 = 45°$。数据序列如图3.3(a)所示，幅度和相位谱分别在图3.3(b)和图3.3(c)中给出。幅度谱的度量是伏特。DFT和傅里叶变换在3.2节介绍，其相互关系为$F(\mathrm{j}\omega) = TX(k)$。用于加速计算的时域抽取的快速傅里叶变换（FFT）算法在3.5节已经介绍了，3.5.1节给出了计算上面序列{1, 0, 0, 1}的DFT的一个例子。

如果待研究的波形长于具有恒定统计矩的时间间隔，则频谱估计可能是不精确的。这也可能是在波形中存在大的噪声分量的情况。这时希望将估计的频谱平滑以得到更好的估计效果。频谱平滑的目的实质上是消除随机性。随机波形的信噪比可以通过波形平均来提高，如果K个波形被平均，则信噪比提高因子为$\sqrt{K}$。因此一种提高频谱估计精确度的方法是将数据分成K个等长度的部分，先确定每一部分的频谱，再将这些频谱平均。这样获得了并画出K个频谱的每个谐波频率分量的平均幅度和平均相位，以产生平均幅度和相位谱。频谱精度可以依据它们的方差来确定。例如功率谱密度的方差越小，估计越精确。因此弄清频谱估计方法对频谱方差的影响是至为重要的。平均法获得频谱估计在11.3.2.1节讨论，还会阐明将K个数据部分的频谱平均以获得频谱估计比直接计算频谱具有更小的方差，方差减小正比于所得数据部分的个数。即使波形所含的噪声成分很少，即高信噪比，这种把K部分结果平均的修正周期图法仍能显著提高精度。然而，将数据分成小块导致每个FFT的数据个数减少，显然造成频谱的粗糙。这个缺点可以通过增加零值来克服（参见11.3.1.2节）。因此我们需要牢记估计精度与谱平滑性的要求是互相对立的，设计中要权衡利弊。另一种平滑周期图的方法是计算加窗数据自相关函数的DFT。这即是11.3.5节介绍的布莱克曼－图基法。由于数据的自相关函数包含着不同延时的数据间相乘后求和平均的计算（参见5.2节），因此同样提高了信噪比（参见5.2.2节）。布莱克曼－图基法得到的品质因数要比修正周期图法大。

数据窗对频谱产生了平滑的作用。特别是如果窗函数的频域具有较小的旁瓣，则主瓣以外的频域数据被有效地滤除，从而得到更好的平滑效果。实际上这种类型的谱平滑有时通过数据谱与选定窗函数的卷积来实现。

参数法及现代频谱估计法在 11.4 节讨论。这些方法不像非参数法那样有一致的算法，必须针对具体情况加以单独讨论。根据参数模型和由这些模型获得的线性系统频响函数频谱来构建数据模型的概念称为现代频谱估计法。

11.3 传统方法

在11.3.1节详细介绍了频谱分析中常见的各种隐患，包括怎样克服它们。加窗技术和数据窗的性质在 11.3.2 节介绍。频谱和谱平滑的一些性质在 11.3.3 节说明。

11.3.1 缺陷

11.3.1.1 抽样率和混叠

在进行频谱分析前，首先要将模拟信号通过抗混叠滤波器，它的作用是在模/数转换前防止抽样信号频谱的混叠。由于不适当的抗混叠滤波器以及抽样率太低，产生的伪低频份量导致了频谱混叠。这个问题在2.2节已经进行了详细的讨论和解释。

11.3.1.2 扇形损失或栅栏效应

如3.2节描述的，离散傅里叶变换DFT包含了频率上均匀分布的谐波幅度和相位分量。随着抽样波形的长度增加，谱线的频率间距减小。假如有个信号分量恰巧落在频谱上两个相邻谐波频率分量之间，则它就不能被正常表达。它的能量由临近的谐波所“瓜分”，而附近的谱“幅度”会失真。均匀谱密度信号的幅度密度谱如图11.1所示。注意谐波频率上的主瓣宽度是有限的，所以在非谐波频率f_{nh}上的信号分量不能被正常表达。解决该困难的方法是使谐波分量频率间距尽可能小，且和信号频率一致。这可以通过在真实数据上加多余的零值数据来获得。增加的零称为增补零值，用以在不附加多余信息的前提下使估计频谱更忠实于真实频谱。足够数量的增补零值N'加到N个数据上，同时满足以下要求：

$$N + N' = 2^m \tag{11.8}$$

从而进行一个基-2的快速傅里叶变换FFT算法，其中m是一个整数。频率$1/(N+N'-1)T$的谐波同信号频率相符，T是抽样间隔。

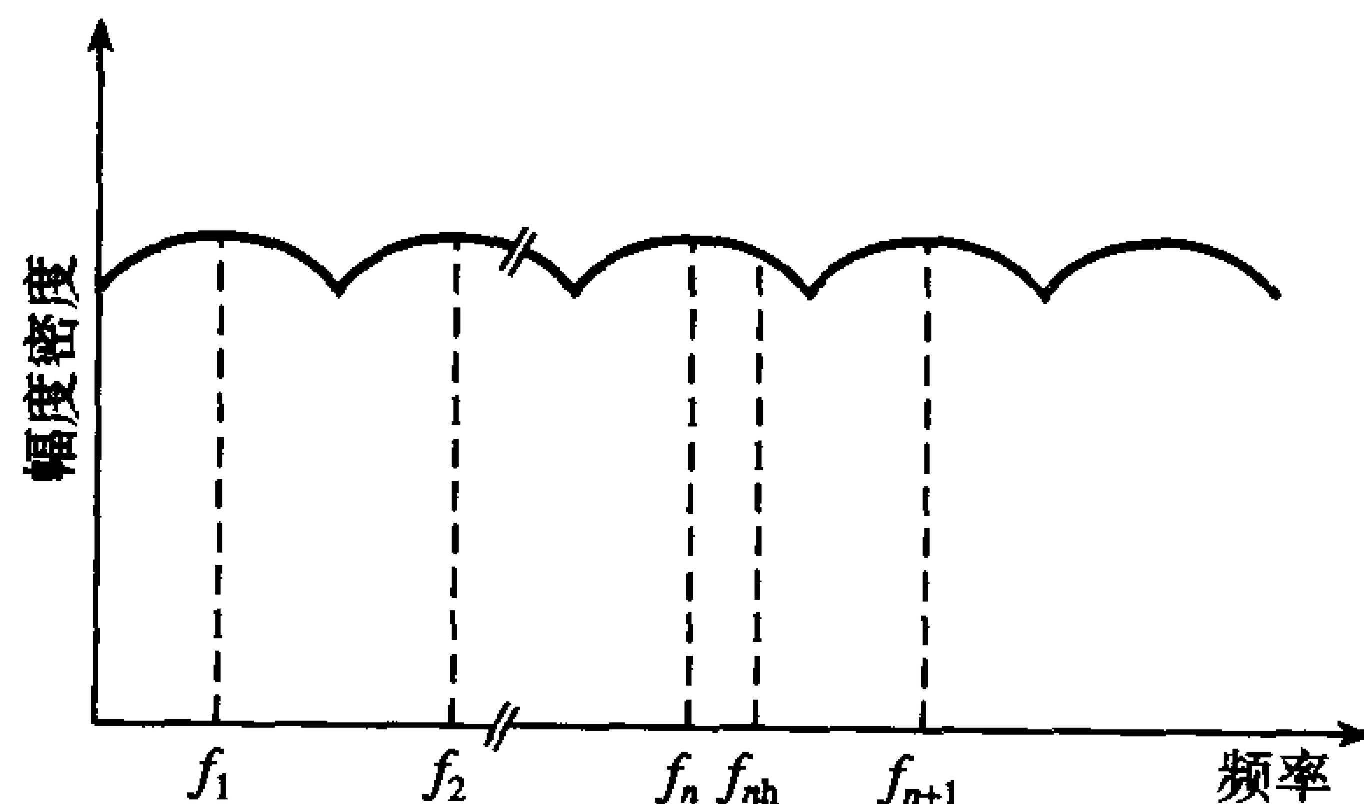


图 11.1 一个均匀谱密度信号的幅度密度谱

扇形损失 SL 定义为恰好在谐波频率中间处的最大处理增益损失（Harris, 1978）:

$$\mathrm{SL}=\frac{|W(\omega_s/2N)|}{W(0)}=\frac{\left|\sum_{n=0}^{N-1}w(nT)\exp(-\mathrm{j}\pi n/N)\right|}{\sum_{n=0}^{N-1}w(nT)} \tag{11.9}$$

这里 W 代表窗函数的DFT（参见11.3.2节），$\omega_s=2\pi f_s$ 是抽样角频率，f_s 是抽样频率，N 是数据个数，n 是数据的标号，$w(nT)$ 是时域抽样的窗函数。

如前所述，有限数据长度的后果是限制可能达到的频率分辨率为 $1/(N-1)T$ (Hz)。这导致产生一个较粗糙的谱，可以通过增补零值使其平滑和连续化。该过程实际只是对谱线上相邻谐波的简单内插过程。分辨率的真正提高必须有更长的数据实现。在增补了 N' 个零值后，谱线的频率间隔变成了 $1/(N+N'-1)T$ (Hz)。

11.3.1.3　趋势消除

数据中的任何趋势必须在计算谱之间被消除，否则数据的趋势会被积累而造成估计谱中很大的误差项。

11.3.1.4　频谱泄漏和频谱混淆

一组抽样数据的FFT并不是数据取自过程的真实FFT。这是由于过程是连续的，而数据是过程的一个现实的抽样值，且被截去了头部和尾部。表示一段长度为 T_s(s)信号的数据是将所有 T_s 间隔内的抽样值乘以1，而所有间隔以外的值都乘以零的结果。这等效于用一个宽度为 T_s、高度为1的矩形脉冲或窗，乘以信号或对其加窗。在本例中，抽样数据 $v(n)$ 由数据值 $s(n)$ 与窗函数值 $w(n)$ 的乘积给出：

$$v(n)=w(n)s(n) \tag{11.10}$$

这里时域乘积等效于频域的卷积，请参见3.3节和5.3节。由此FFT值的第 n 个谐波为

$$V(\omega_n)=\sum_{k=-N}^{N}W(\omega_n-\omega_k)S(\omega_k) \tag{11.11}$$

其中 ω_n 是第 n 个谐波的角频率，$V(\omega_n)$ 是频点 ω_n 处的复DFT分量，$W(\omega)$ 是频点 ω_n 处窗函数的DFT，$S(\omega_k)$ 是频点 ω_k 处信号的真实DFT分量。

从11.11式可以看出，计算出的频谱包含着数据的真实频谱与窗函数的卷积。矩形脉冲 $S_R(\omega_n)$ 的幅度谱由下式给出，即所谓的狄拉克（Dirichlet）核：

$$S_R(\omega_n)=\frac{T_s\sin(\omega_n T_s/2)}{\omega_n T_s/2}=\mathrm{Sa}\left(\frac{\omega_n T_s}{2}\right) \tag{11.12}$$

$\mathrm{Sa}(\omega_n T_s/2)$ 是 $\omega_n/2$ 的抽样函数（参见3.1.1节和3.1.2节），如图3.2(a)所示。它分别包含了一个峰值点频率为0的主瓣和无数峰值点频率为 $(n+0.5)/T_s$ 的旁瓣。现在信号在频点 f_n 处的一个简单正弦波分量的幅度谱包含了在频点 $\pm f_n$ 处的两个冲激。与抽样函数的卷积后产生了如图11.2的谱。两个冲激已被转变成两个重叠的抽样函数。由于矩形窗的旁瓣作用，计算出的频谱产生了伪峰。这信号的每个频率分量都是同样起作用的，所以在增加或消去大量窗的旁瓣或主瓣时，信号的幅度谱会产生失真。这种称为频谱泄漏的效应会导致伪峰或消去谱中真实的峰。为了避免此种效应，需要使用能降低旁瓣的特殊形状窗来与数据相乘。合适的窗在中点处的值为1，到两端 $n=0$ 和 $n=N-1$ 处逐渐减小到0。现在已至少提出了23个这样的窗，哈里斯（Harris, 1978）研究了它们的相互适宜性。

为了最小化频谱泄漏，应选择能最小化旁瓣的窗形状。遗憾的是，它会增大主瓣宽度，导致它进入临近的旁瓣，使旁瓣混淆。在每个谐波频率重复后，总的后果是信号频谱的混淆。因此必须小心地选择窗的种类和参数，使其在频率分辨率和谱估计的统计精度之间找到一种平衡。

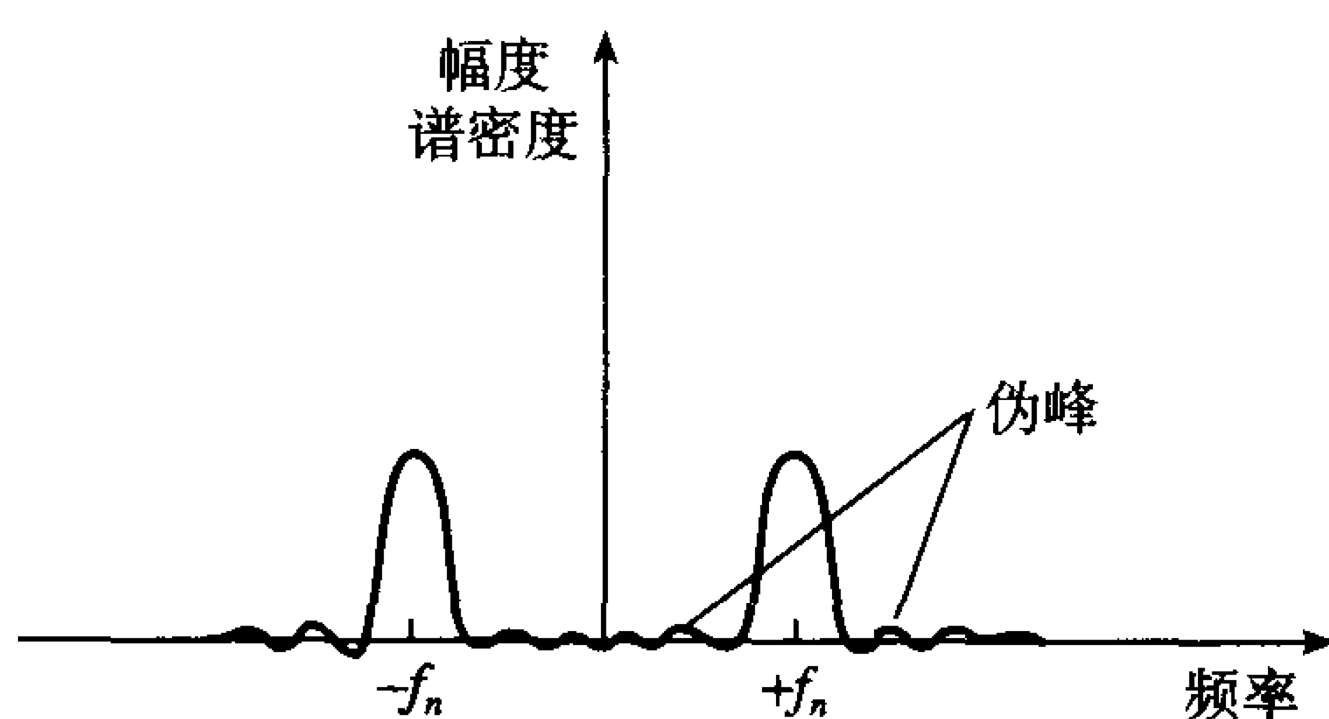


图 11.2 一个正弦信号与抽样函数卷积后的幅度谱密度

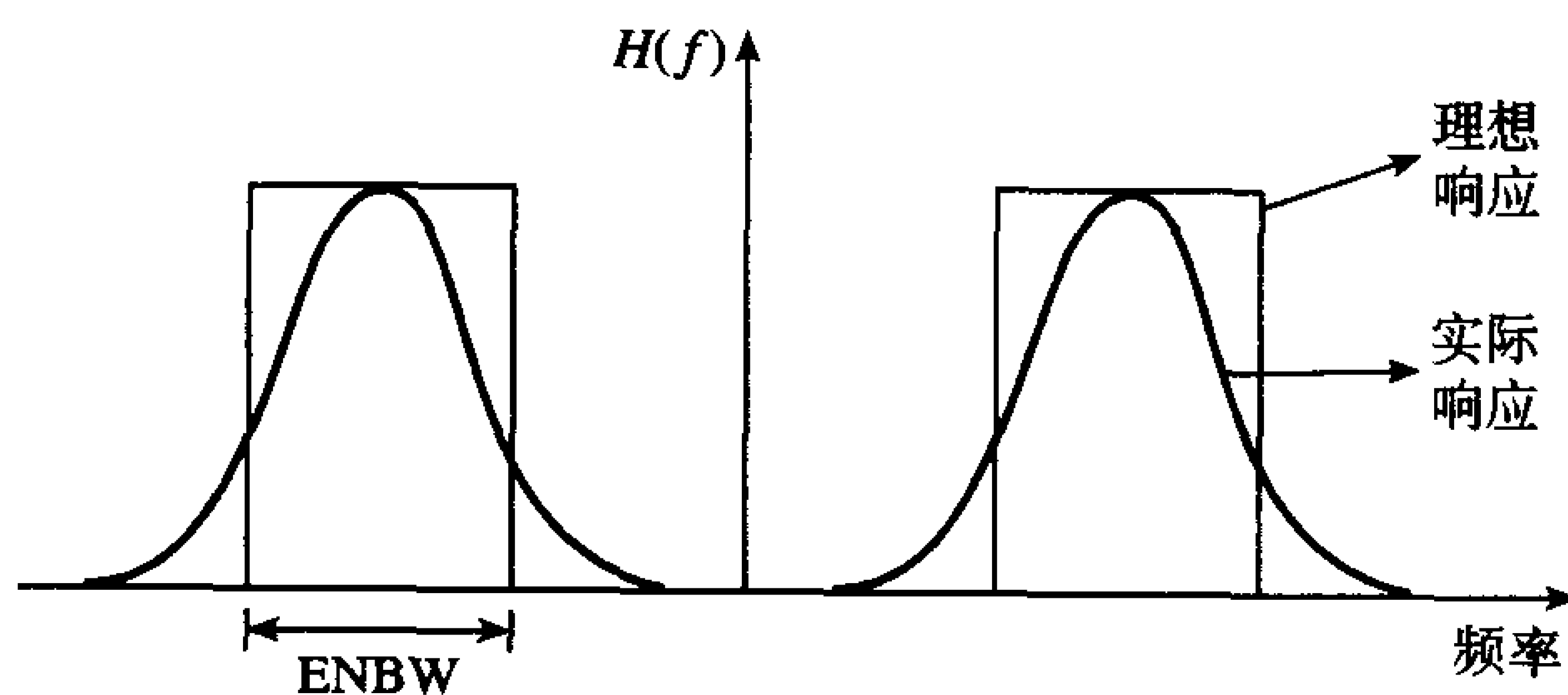
11.3.2 加窗

在本节介绍不同性质的窗，基本上是在时域。然而，再一次强调加窗既可以在时域（数据窗）也可以在频域（频率窗）进行，因为时域相乘就等效于频域的卷积（参见 11.3.1.4 节）。频域加窗因此可以通过频域窗与信号频谱的卷积来实现。该过程可以应用 5.104 式获得。

11.3.2.1 窗的性质

等效噪声带宽

在 11.3.1.4 节看到，由于频谱泄漏现象，理论上代表幅度谱密度的冲激函数变成了抽样函数。即无限窄的频谱分量被具有较大带宽的抽样函数所取代。这些导致信号分量偏置的函数旁瓣可以看成给信号贡献了不需要的噪声，而频域窗则是一个宽带滤波器。从这个观点来看，希望设计降低旁瓣幅度的低噪声带宽窗。为了相互比较的目的，按照等效噪声带宽来测量不同种类窗的噪声带宽。等效噪声带宽被定义为一个理想矩形滤波器的带宽，它与讨论中的谱滤波器能通过同量的白噪声（参见图 11.3）。使用这种定义通过比较等效噪声带宽来比较不同窗的旁瓣性质。因此等效噪声带宽是一个重要的窗参数。从这个观点看它越小窗就越好。



理想响应曲线下的面积 = 实际响应曲线下的面积

图 11.3 一个滤波器的等效噪声带宽

等效噪声带宽由下式给出：

$$\mathrm{ENBW}=\frac{\sum_{n=0}^{n=N-1} w^2(nT)}{\left[\sum_{n=0}^{n=N-1} w(nT)\right]^2} \tag{11.13}$$

重叠相关

当数据被加窗时，数据序列的端部将缩减至零，这说明信息的损失。特殊情况下，在缩减区发生的短持续时间事件会丢失。解决该问题的一种方法是将数据重叠分块，再分别加窗和变换每一部

分。如果重叠大约为 50% 或 75%，则数据的大多数特征会保留下来。将各部分谱进行平均来获得真实频谱的估计。数据的分块如图 11.4 所示。该过程称为冗余或重叠处理。一般 50%~75% 的重叠处理对大多数权重函数能提供至多 90% 的性能提升（DeFatta et al., 1988）。通过将各个部分的谱平均，频谱的方差被降低。对 K 个具有同样统计特性但相互独立的样本，平均后的谱方差是原来各自谱方差的 $1/K$。然而，这对重叠分块的谱平均来说是不正确的，因为它们互相相关。对于 50% 和 75% 的重叠情况，其谱平均与各自谱的方差在哈里斯（1978）的著作中给出，从中可以看到，例如四个谱平均后方差降低到原来谱方差的 25%。这说明频谱估计有了显著的提高。

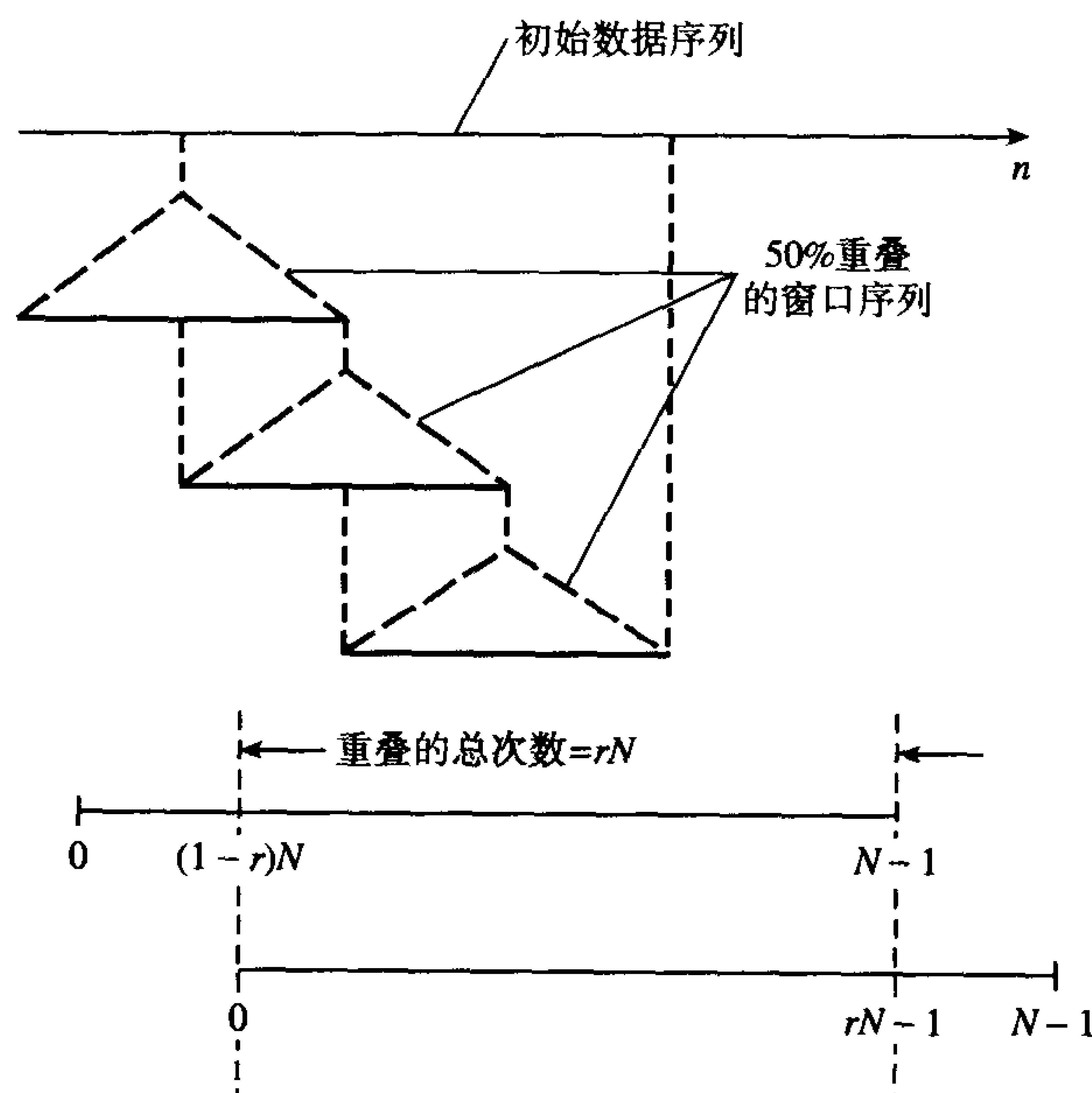


图 11.4　用于重叠处理的数据分块

处理增益

处理增益（PG）定义为加窗后的输出信噪比与加窗前的输入信噪比的比值：

$$PG = \frac{(S/N)_{O/P}}{(S/N)_{I/P}} \tag{11.14}$$

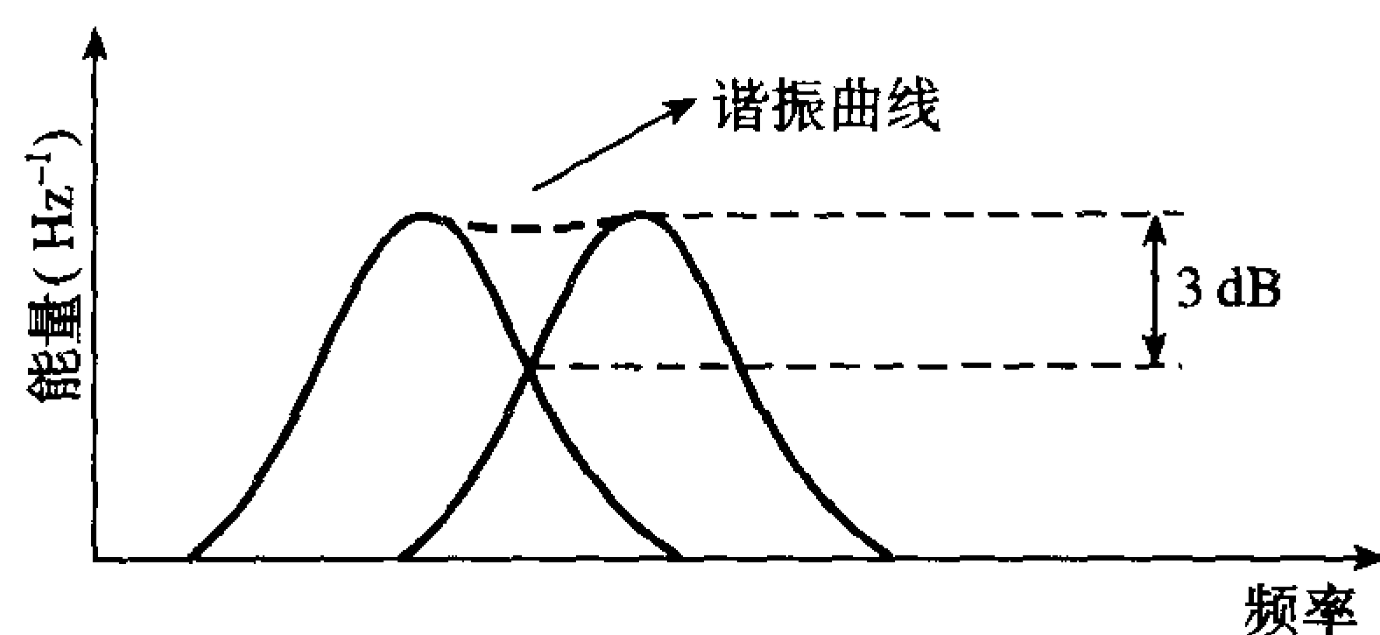
处理增益的大小取决于窗的形状，因为窗确定了等效噪声带宽（参见前面有关的部分）。窗（两端）的缩减降低了信号功率，带来了处理损失（PL），而旁瓣的存在则增加了噪声带宽。

最坏处理损失

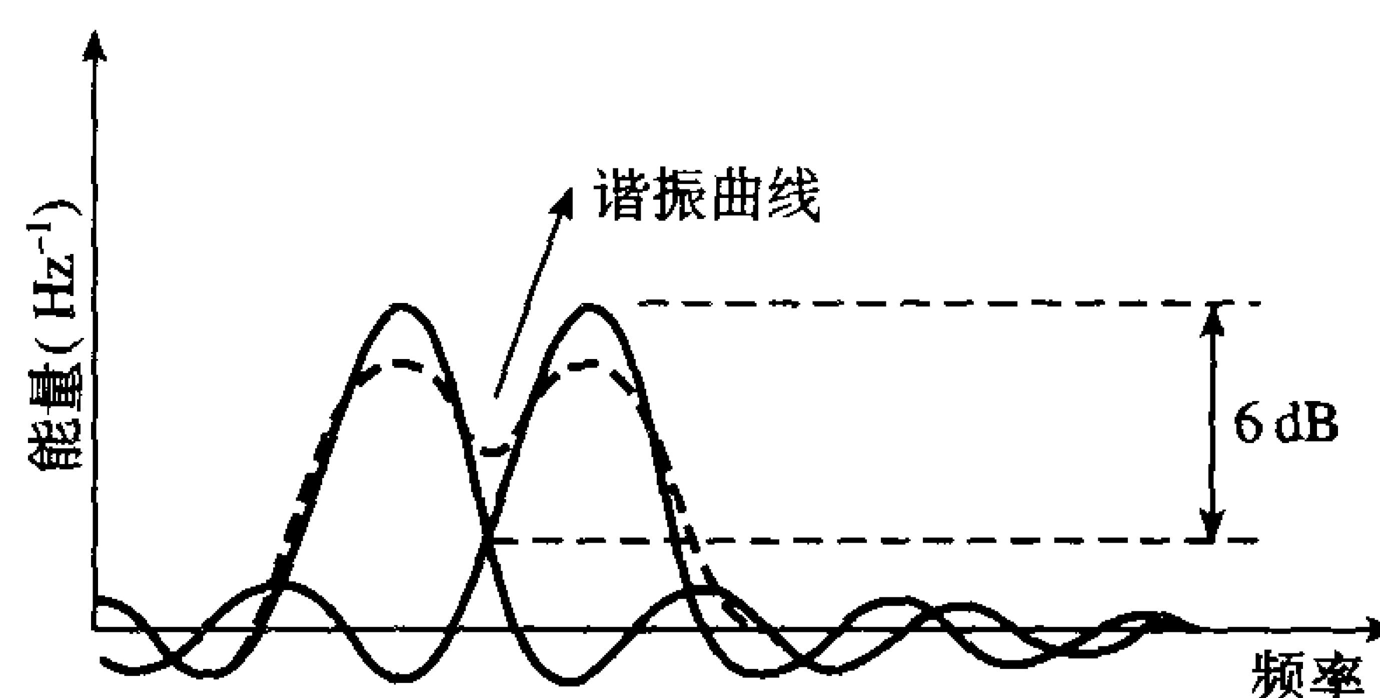
最坏条件下的处理损失（WCPL）定义为一个窗的最大扇形损失加上（在 dB 意义上）它的处理损失。它代表由于加窗导致的输出信噪比降低和最坏条件的频率定位。一般来说它的数值在 3.0 ~ 4.3 dB 之间。对于该数值超过 3.8 dB 的窗应避免使用。这包括了矩形窗、泊松窗（$\alpha = 4$）、汉宁 - 泊松窗（$\alpha = 2.0$）、柯西（Cauchy）窗（$\alpha > 4$）及最小四抽样点的布莱克曼 - 哈里斯窗。

最小分辨率带宽

两个完全相同的谱峰重叠在通常情况下能够分辨开的条件是它们的重叠没有超过各自的 3 dB 点（参见图 11.5(a)）。然而，如果频谱是 DFT 得到的，相邻谱分量被窗所加权再相关积累，即旁瓣也被加入进去。在交叉处每个分量的增益必须不超过 0.5。这说明谱分辨率是由分量的 6 dB 带宽来确定，而不是 3 dB 带宽（参见图 11.5(b)）。



(a) 谱峰分辨取决于3 dB带宽



(b) DFT的谱分辨取决于6 dB带宽

图 11.5　最小分辨带宽

数据窗的偏差效应

数据与一个缩减窗相乘降低了缩减处的抽样点幅度，从而降低了信号总功率。可以看出每个频率分量受到窗的同样影响，且乘积因子正比于相关信号增益的平方根，它代表了数据窗相对于一个电压波形的归一化功率。因此信号功率的降低可以在不产生功率密度谱失真的情况下还原。窗还造成数据均值的变化，明显增强了谱中低频分量的能量。因此需要采用一些补偿它的方法，但从加窗数据中直接减去均值会导致显著的高频旁瓣。

图 11.6 显示了 64 个数据点的能量密度谱，数据的均值先被减去，然后再乘以一个凯塞－贝塞尔窗的相应值。可以看到谱包含了由于加窗处理带来的低频分量。图11.7显示出加窗数据均值被预先减去的谱。可以看到低频分量已经消除，但在高频处存在显著的旁瓣结构。下面的例子给出了怎样克服加窗带来的不利影响。

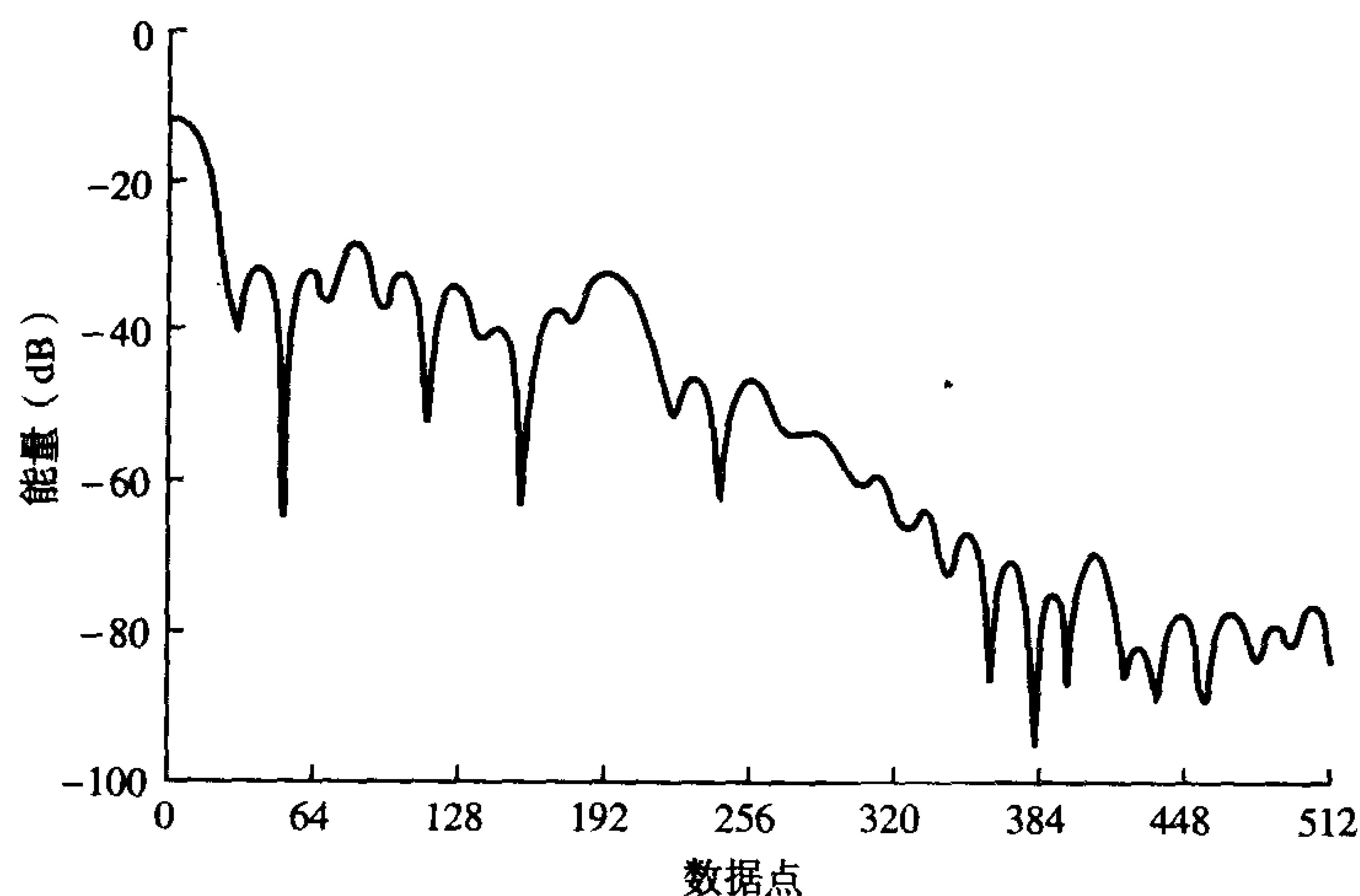


图 11.6　1 s 的刺激间隔（ISI）CNV 的能量谱，凯塞－贝塞尔窗，64 个数据点，1024 点 FFT

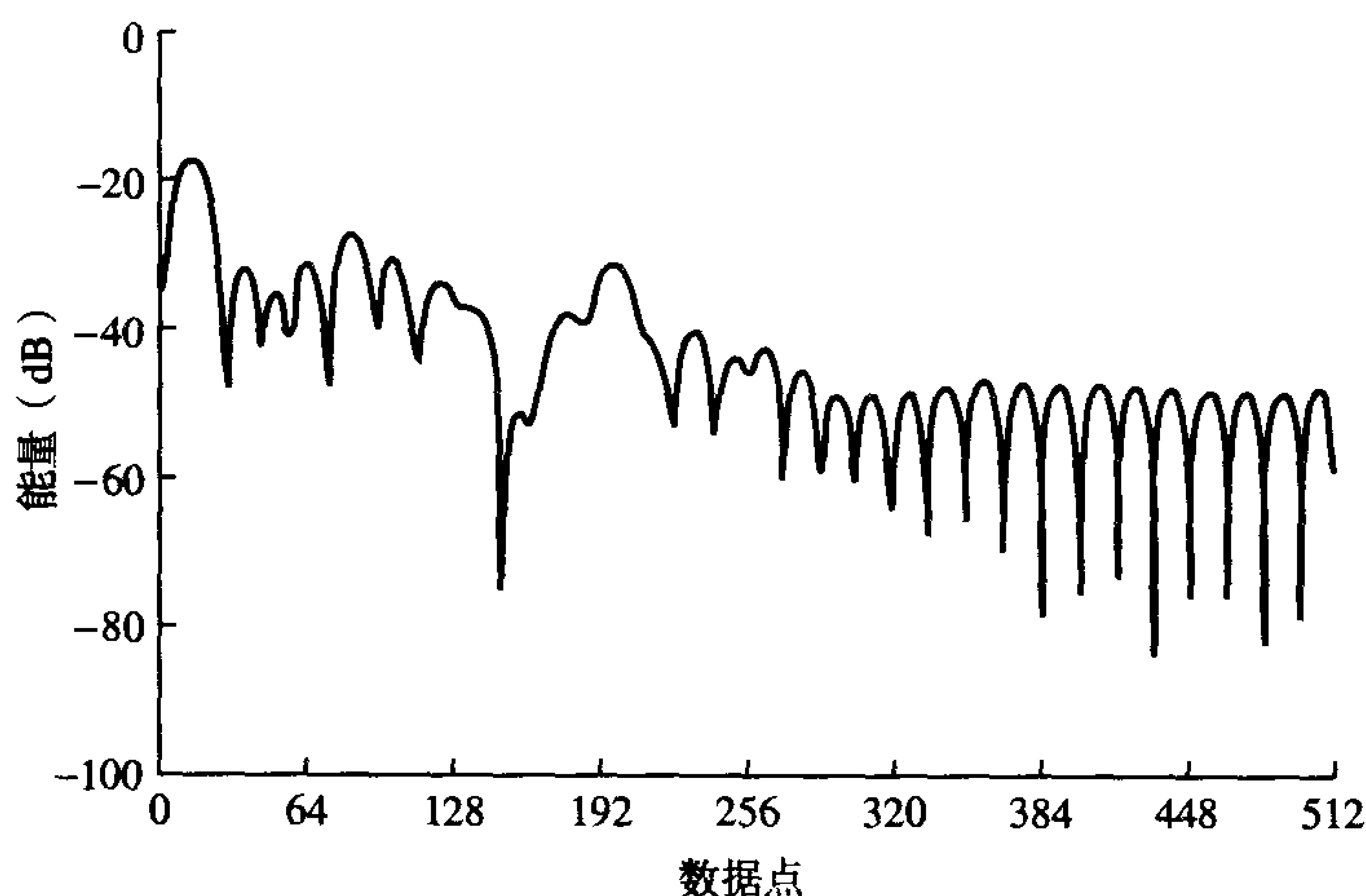


图 11.7　1 s的ISI CNV的能量谱，凯塞－贝塞尔窗，64个数据点，1024点FFT。数据的均值被预先减去

例 11.1　阐述加窗带来的双重效应——降低信号能量在谱中引入低频分量，可以通过对数据的线性函数加窗而不是对数据本身加窗来克服（Jervis et al., 1989a）。

解：

假定初始数据$s(n)$是零均值的。由于加窗带来的均值水平通过从$s(n)$中减去一个常数k_1来消除。新的加窗数据是$s^1(n)$，且

$$s^1(n) = w(n)[s(n) - k_1] \tag{11.15}$$

其中$w(n)$是窗函数值或权重。加窗导致的信号能量降低可以通过将$s^1(n)$的每个值与一个精心挑选的常数k_2来恢复。因此数据点变成

$$S(n) = k_2 w(n)[s(n) - k_1] \tag{11.16}$$

k_1所需的数值可以通过要求$S(n)$的均值必须为零来找到。所以

$$\sum_{n=0}^{N-1} S(n) = 0$$

由此

$$k_2\left[\sum_{n=0}^{N-1} w(n)s(n) - \sum_{n=0}^{N-1} w(n)k_1\right] = 0$$

则有

$$k_1 = \frac{\sum_{n=0}^{N-1} w(n)s(n)}{\sum_{n=0}^{N-1} w(n)} \tag{11.17}$$

数据加窗前的归一化交流（ac）能量为

$$E[s\,(n)-k_1)^2] = \sigma_{sN}^2 \tag{11.18}$$

这里E代表求期望值，σ_{sN}^2是$s(n)$的方差，均值为k_1。数据加窗后的归一化交流能量为

$$E\{k_2^2 w^2(n)[s(n) - k_1]^2\}$$

其中 $w(n)$和 $s(n)$互相独立。现在

$$\begin{aligned}E\{k_2^2w^2(n)[s(n)-k_1]^2\} &= E[k_2^2w^2(n)]E\{[s(n)-k_1]^2\} \\ &= k_2^2E[w^2(n)]\sigma_{sN}^2\end{aligned} \quad (11.19)$$

要求 k_2 的取值使得加窗前后数据的能量相同，这可以令 11.18 式和 11.19 式相等而得到：

$$\sigma_{sN}^2 = k_2E[w^2(n)]\sigma_{sN}^2$$

所以

$$k_2^2 = \frac{1}{E[w^2(n)]} = \frac{1}{(1/N)\sum_{n=0}^{N-1}w^2(n)} = \frac{N}{\sum_{n=0}^{N-1}w^2(n)}$$

所以

$$k_2 = \left[\frac{N}{\sum_{n=0}^{N-1}w^2(n)}\right]^{1/2} \quad (11.20)$$

将 11.17 式和 11.20 式代入 11.16 式最后得到

$$S(n) = w(n)\left[s(n) - \frac{\sum_{n=0}^{N-1}w(n)s(n)}{\sum_{n=0}^{N-1}w(n)}\right]\left[\frac{N}{\sum_{n=0}^{N-1}w^2(n)}\right]^{1/2} \quad (11.21)$$

图 11.8 显示了由此得到的加窗数据均值被消除后的能量谱，并应用 11.21 式恢复信号能量。可以看到由于加窗导致的直流分量和旁瓣效应同时消除了。

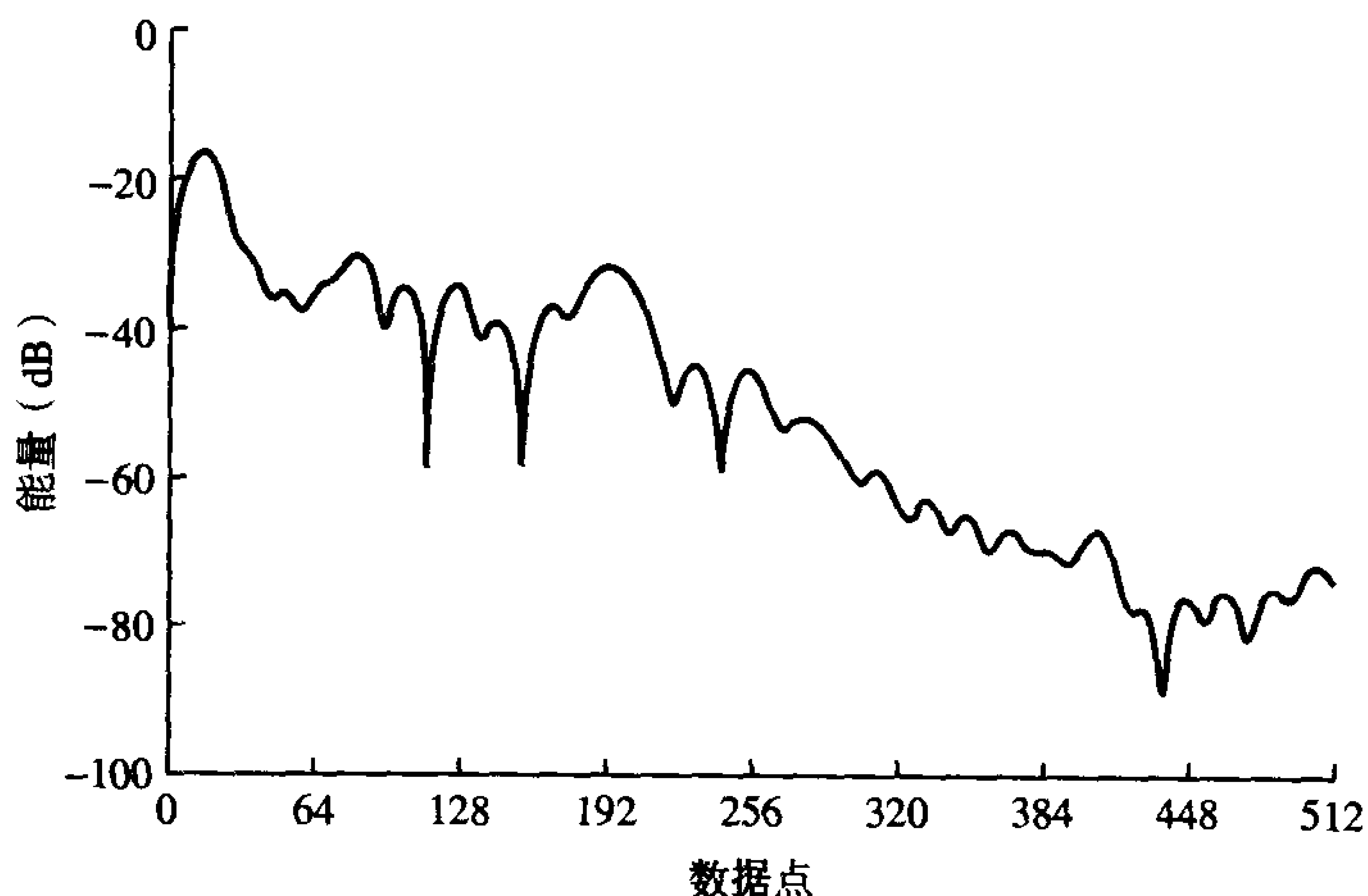


图 11.8　1 s 的 ISI CNV 的能量谱，凯塞－贝塞尔窗，64 个数据点，1024 点 FFT。数据预处理以消除加窗的均值分量和保持平均功率

因此推荐的程序是在计算 DFT 前根据 11.21 式修正数据 $s(n)$。这等效于先从数据中减去 k_1，再在加窗前乘上 k_2。

11.3.2.2　窗的选择

哈里斯（Harris, 1978）曾经详细考虑了不同窗特性对加窗性能的影响，发现对窗质量主要的影响是最高旁瓣水平和最坏处理损失。因此受欢迎的窗有布莱克曼－哈里斯窗、多夫－切比雪夫

(Dolph-Chebyshev)窗和凯塞-贝塞尔窗。著名的图基窗(余弦形式缩减)、泊松窗、汉宁窗和哈明窗的性能均较差。

大多数窗的缩减方式和它们的形状在很多情况下可以通过选择一个参数α的值来调节。它的效果是调整主瓣宽度和旁瓣水平。实际应用中加窗的部分技巧在于通过试验和误差来选择α的值，从而最优化最终结果。

例 11.2　不同窗对一个幅度谱的影响　图 11.9(a)显示了大小两个正弦波的 DFT 分量，它们的幅度相差 40 dB，频率分别在 100f和 120f，这里f是对应于一个长度为T_s(s)的数据记录的第一个谐波频率，该纪录是未加窗的，或者说是使用了矩形窗。当信号与窗（记录）长度存在这种谐波关系时，谱表现为周期性和无限长，即便是矩形窗也能忠实体现原信号的特点。在图 11.9(b)中这个周期性关系被打破了，只是由于将较强信号的频率改成不再是谐波频率的 102.5f。可以看到结果是它的旁瓣水平大幅提高，几乎淹没了较小信号。这种后果可以通过选择一种适宜的窗来克服。图 11.10(a)和图 11.10(b)分别显示了$\alpha=0.1$和$\alpha=0.5$的余弦缩减（图基）窗，而图 11.11(a)和图 11.11(b)则给出了对应的谱。图 11.12(a)和图 11.12(b)分别给出了哈明窗（$\alpha=0.54$）和结果频谱。图 11.13(a)~图 11.13(c)分别给出了$\alpha=2.0$、3.0 和 4.0 的凯塞-贝塞尔窗，同时图 11.14 显示了对应的谱。结果是最接近真实谱的窗是$\alpha=4.0$的凯塞-贝塞尔窗。然而，也要注意到增大α降低旁瓣的同时，主瓣宽度也增加了。需要找到一个折中的α值。当然在实际情况中所有谐波都会受影响，如果想要避免“伪谱”，那么针对多频性能下的窗的选择是非常重要的。

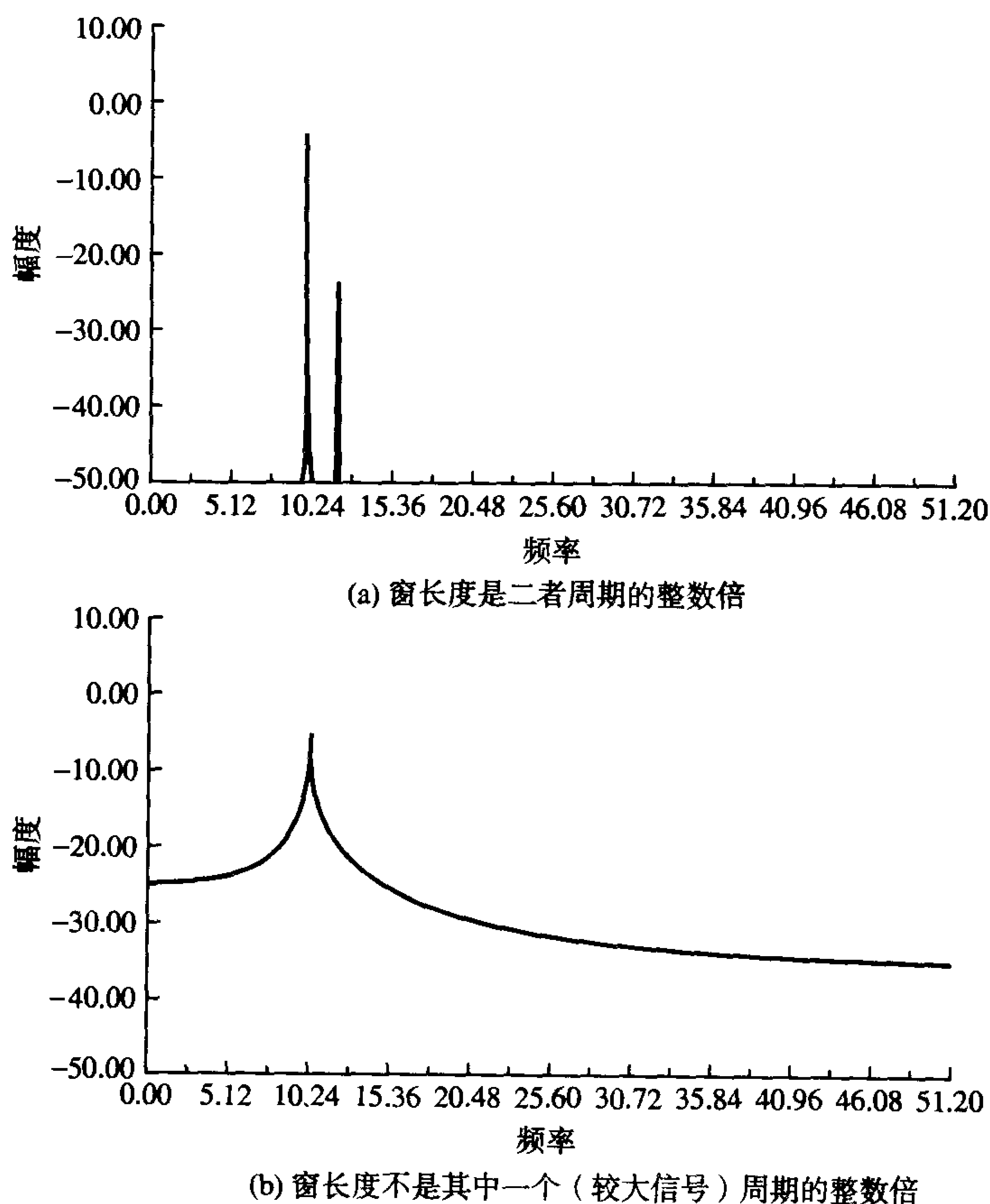


图 11.9　两个幅度相差 40 dB 的正弦波的幅度谱

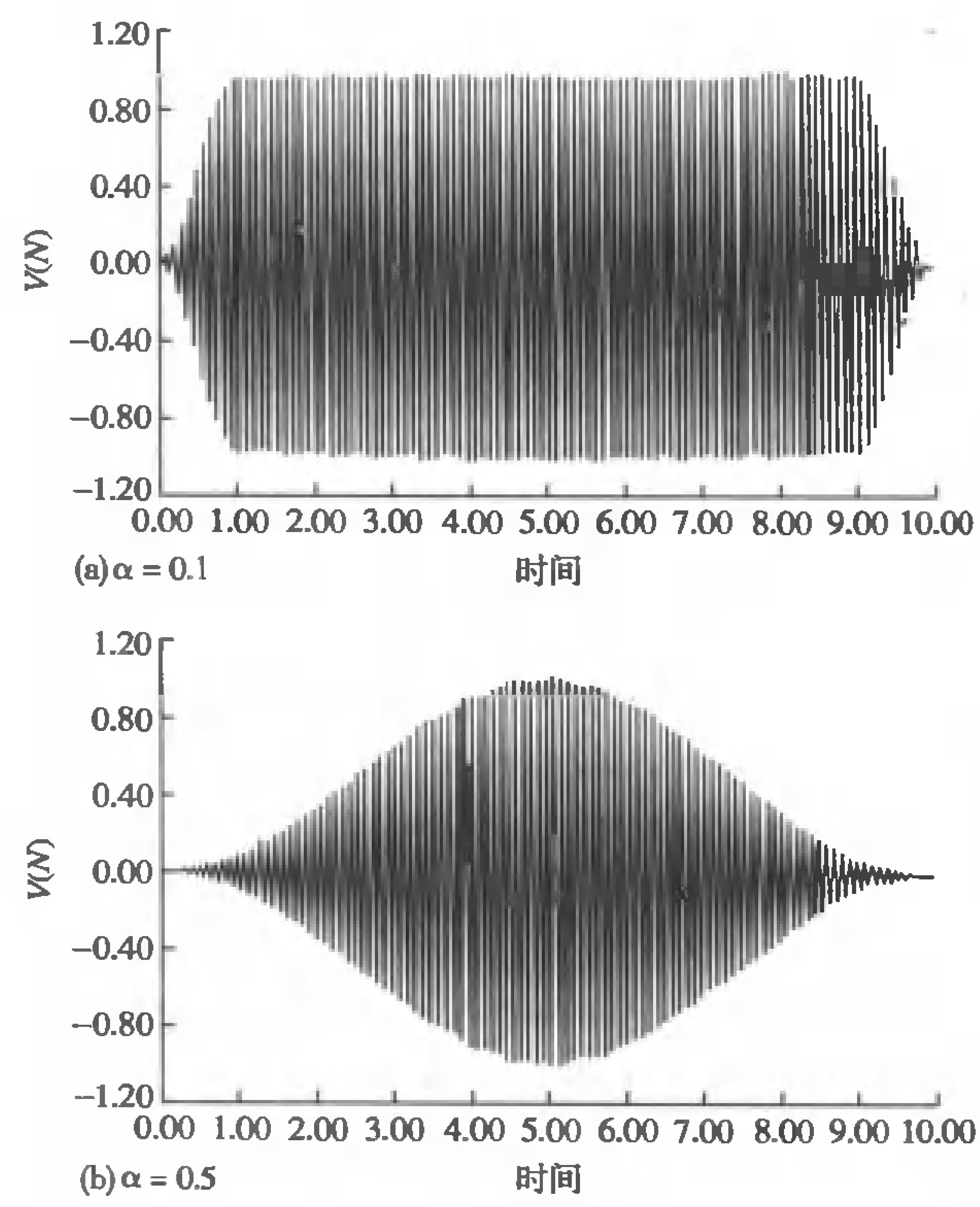


图 11.10　余弦缩减（图基）窗

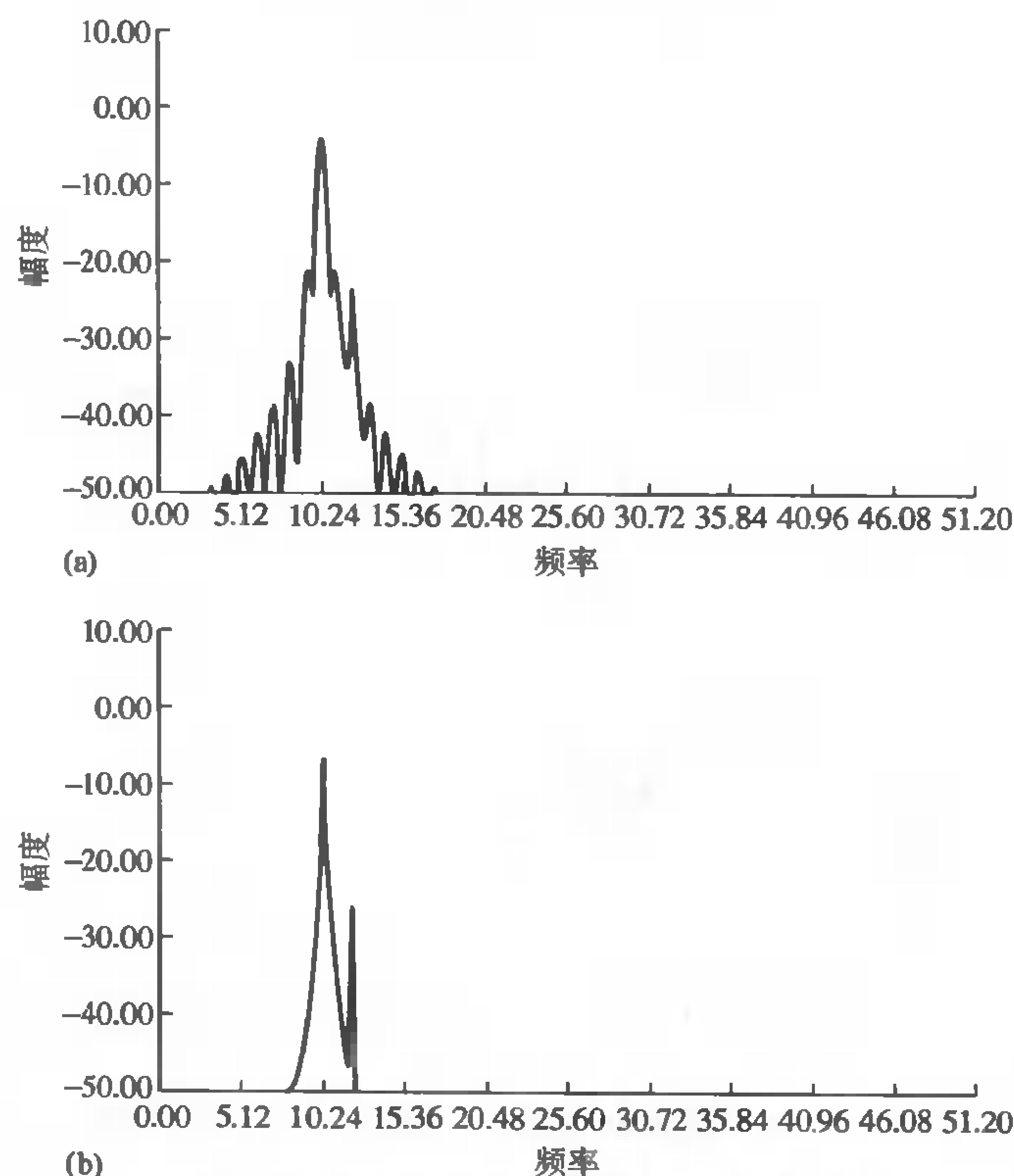
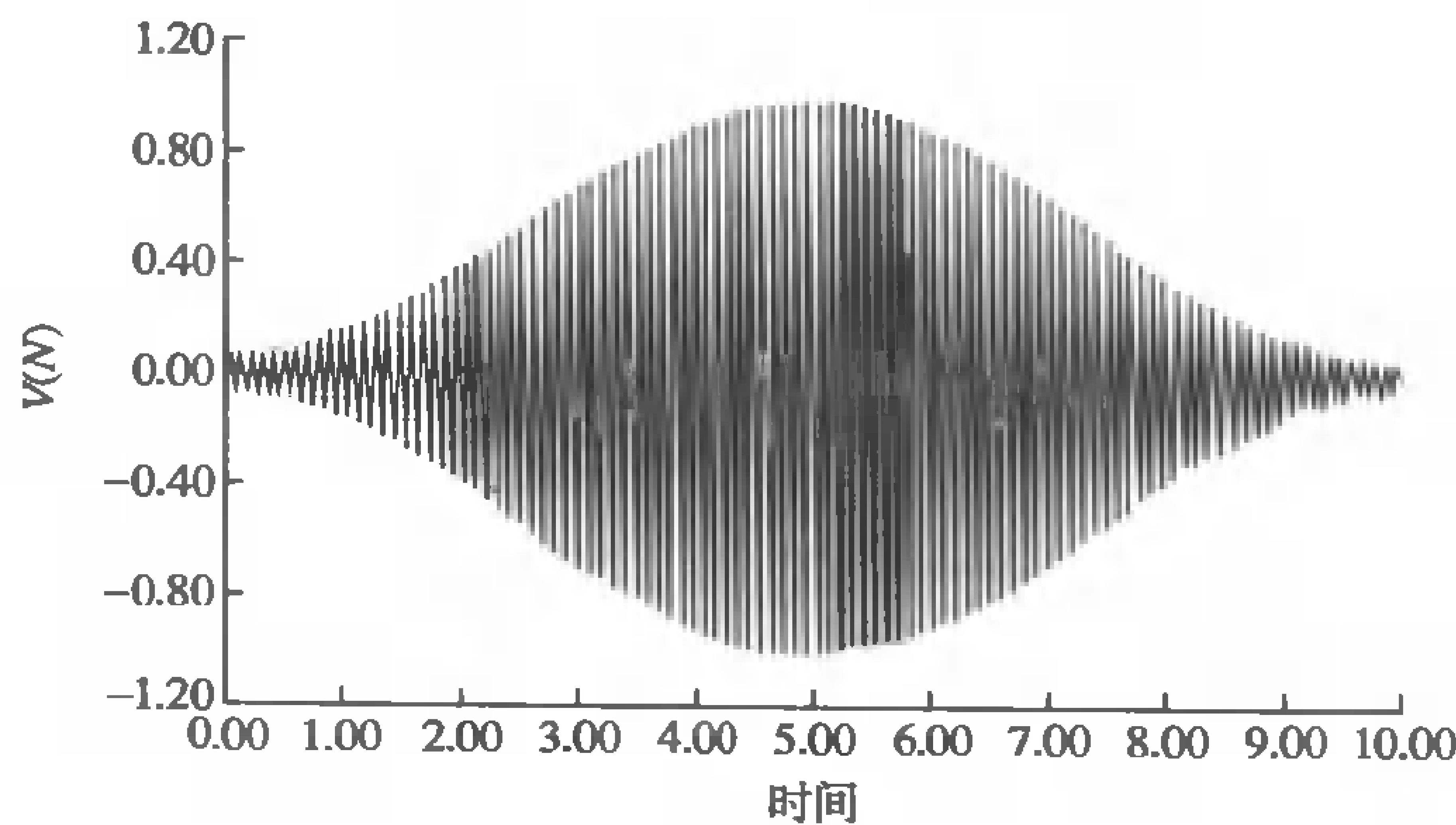
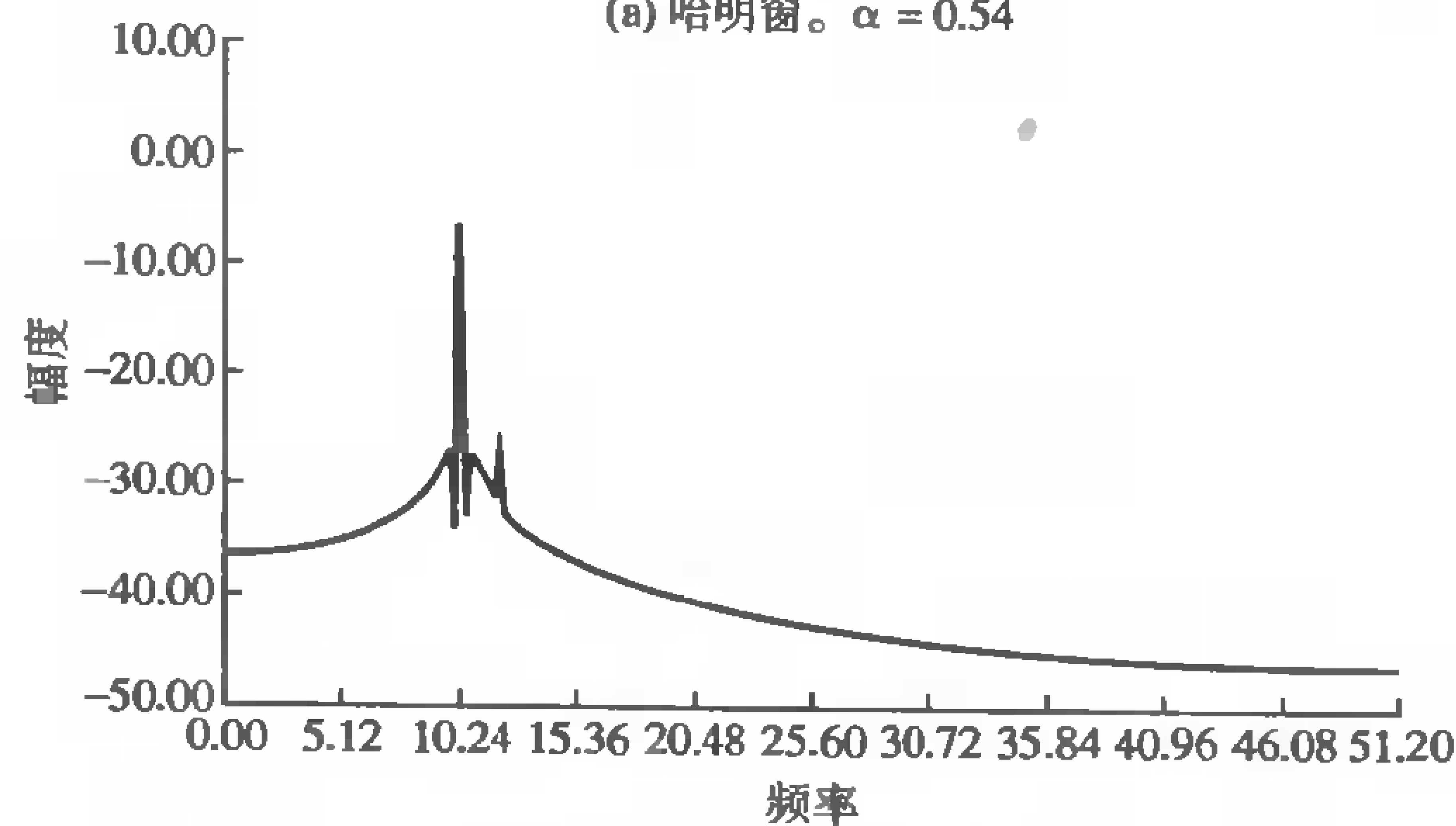


图 11.11　两个正弦波乘以余弦缩减窗后的幅度谱

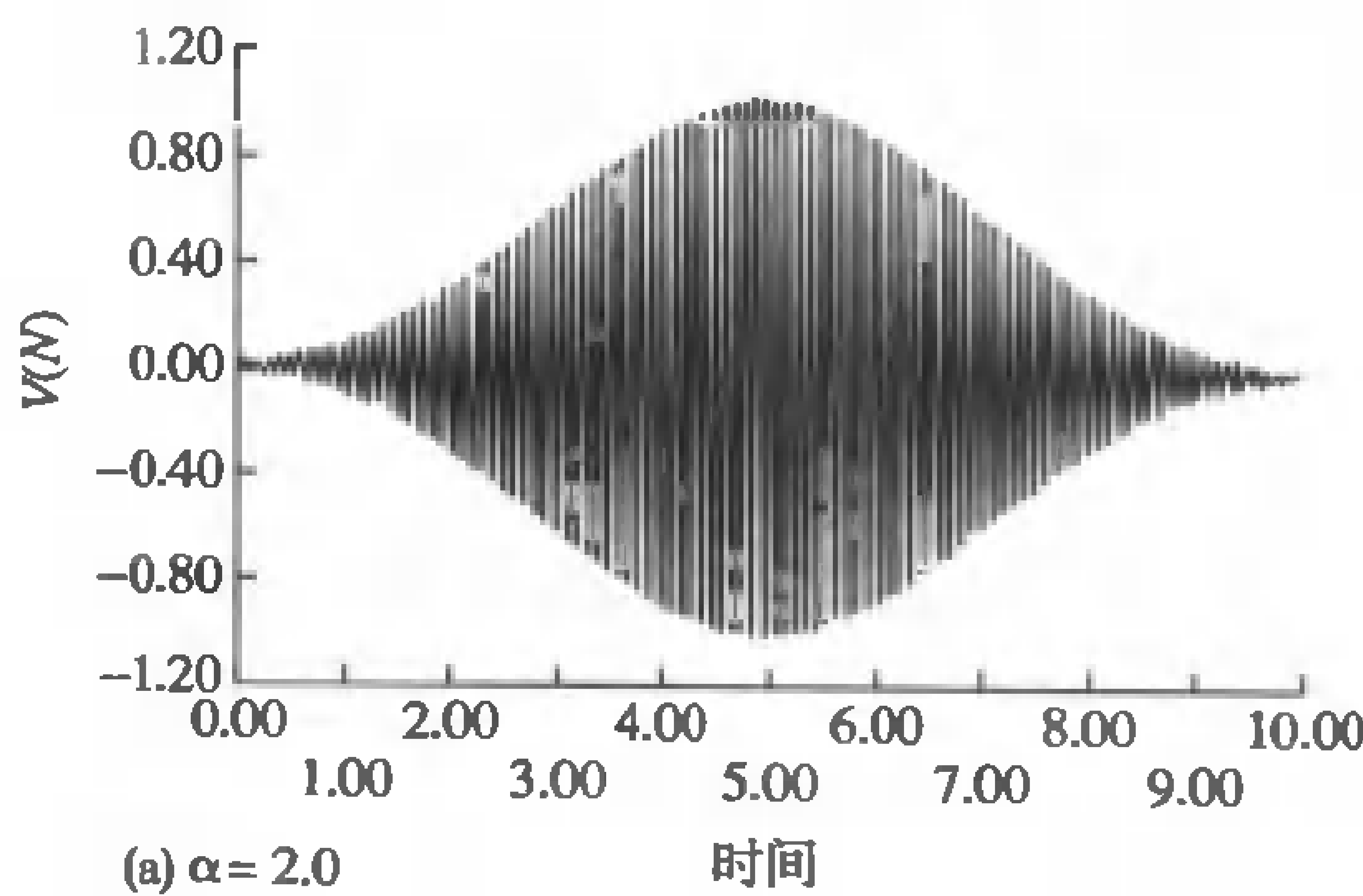


(a) 哈明窗。α = 0.54

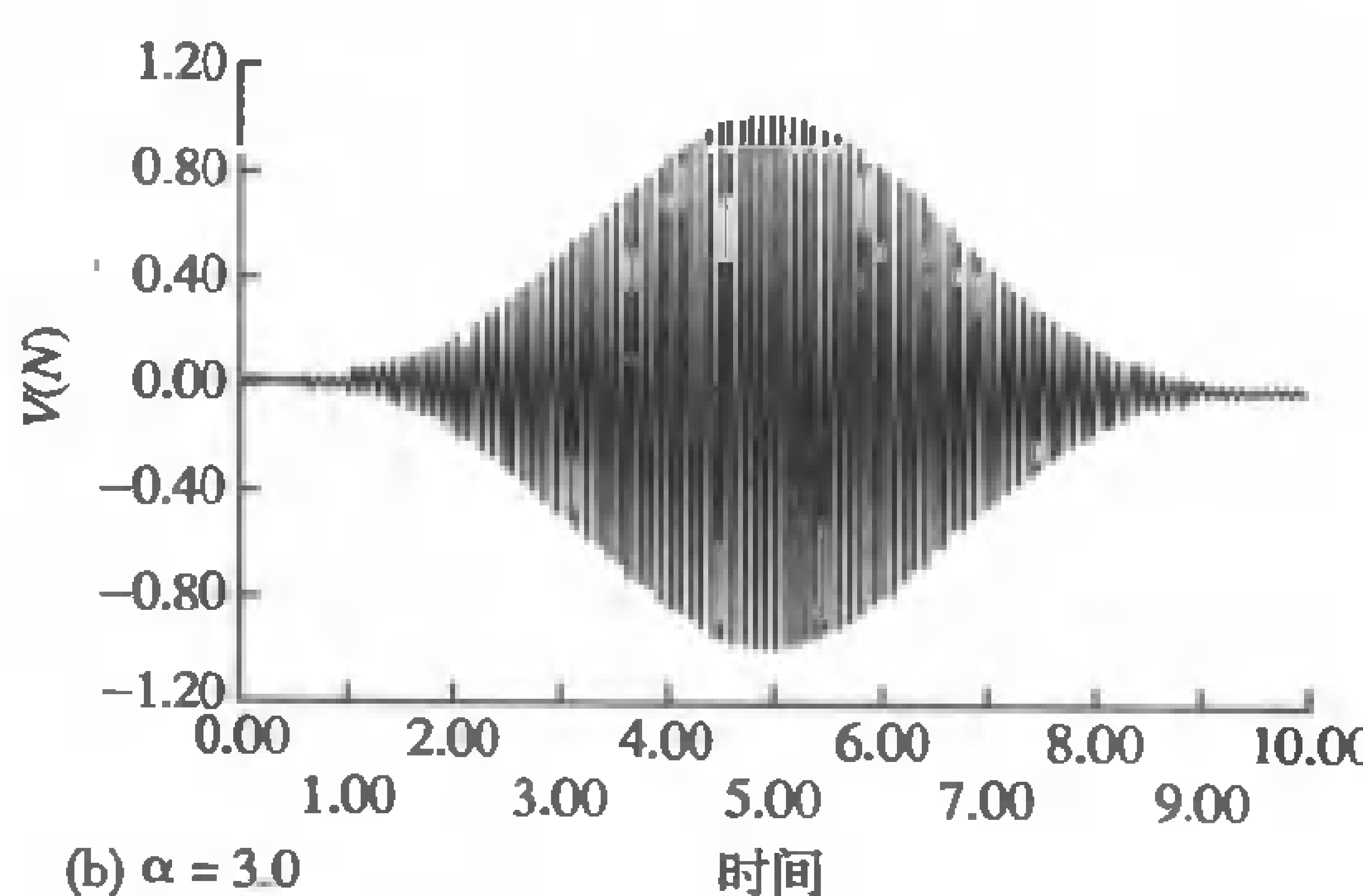


(b) 相应两个正弦波的幅度谱

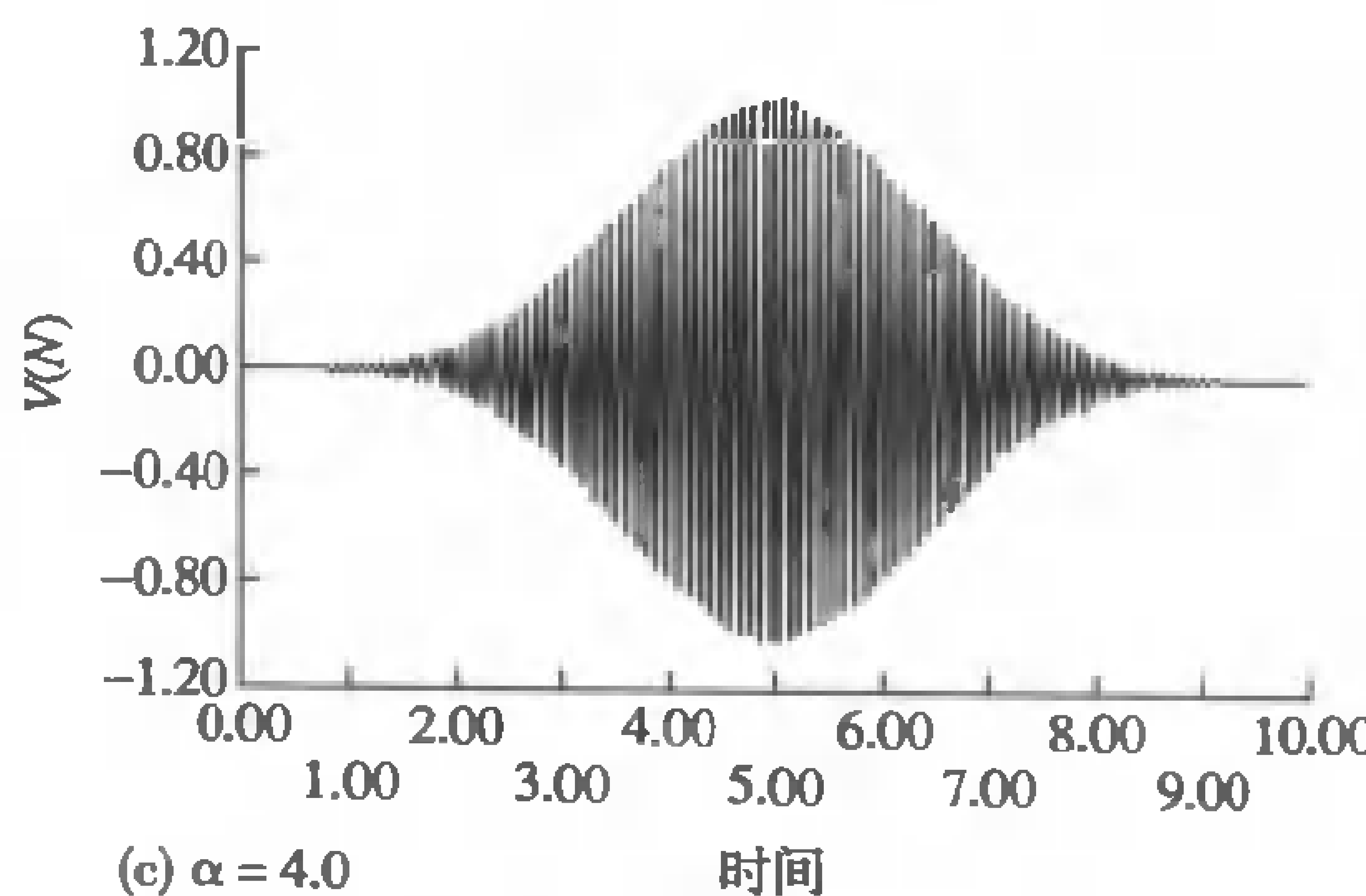
图 11.12　哈明窗和相应两个正弦波的幅度谱



(a) α = 2.0



(b) α = 3.0



(c) α = 4.0

图 11.13　凯塞 – 贝塞尔窗

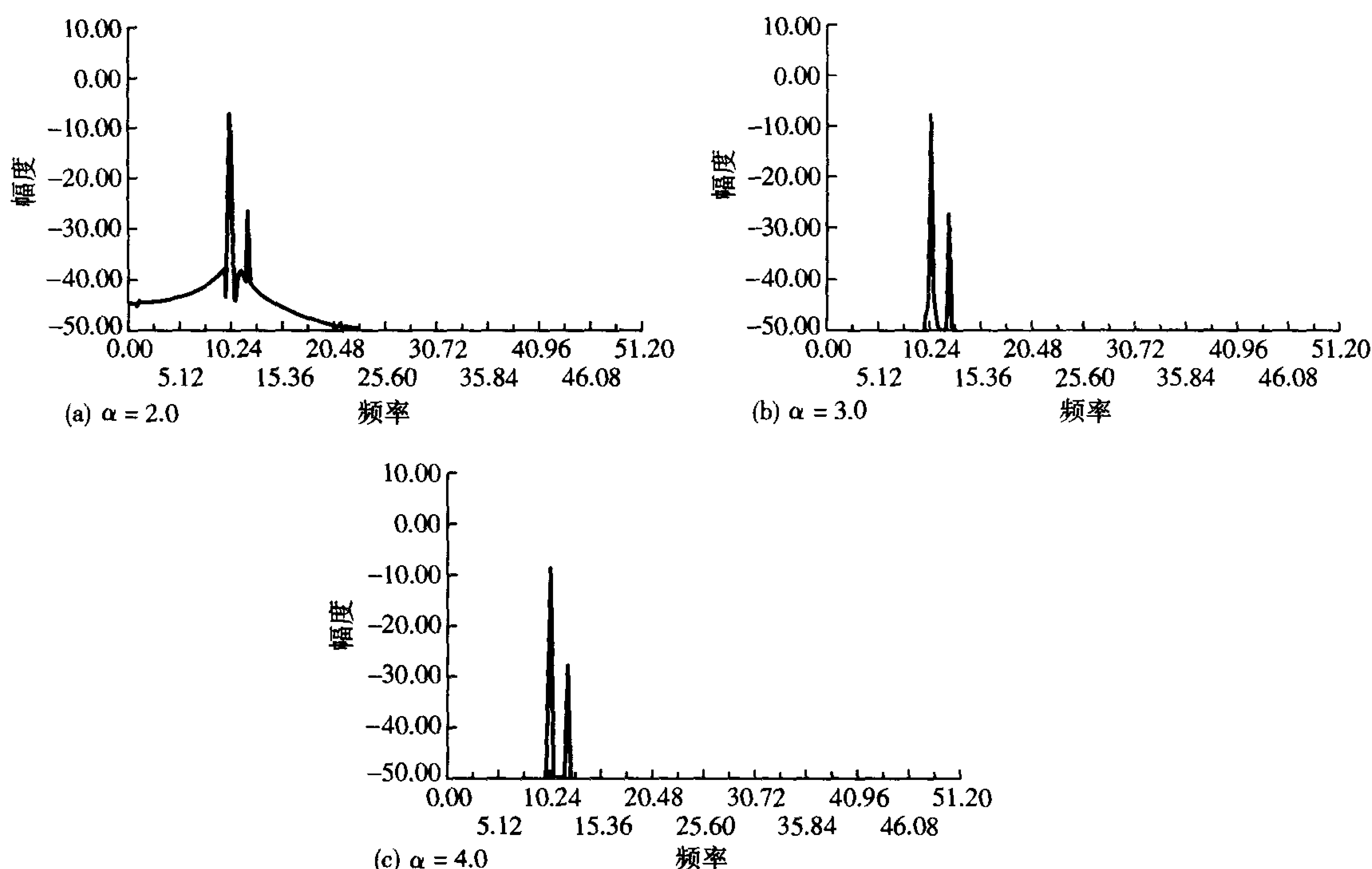


图 11.14 使用图 11.13 中相应的凯塞 - 贝塞尔窗计算得到的两个正弦波的幅度谱

多夫 - 切比雪夫窗给出了在低旁瓣和最坏处理损失方面最好的结果，但是它的旁瓣的相关积累损害了它的多频检测性能。还有，它的旁瓣结构表现出对参数误差的极端敏感。因此布莱克 - 哈里斯窗和凯塞 - 贝塞尔窗是比较受欢迎的。凯塞 - 贝塞尔窗的独特优势在于它的参数很容易产生，而且通过改变 α，旁瓣水平与主瓣宽度的折中很容易实现。凯塞 - 贝塞尔窗的表达式是（Kuo and Kaiser, 1966）

$$w(n_{\mathrm{KB}}) = I_0\left\{\pi\alpha\left[1.0-\left(\frac{n_{\mathrm{KB}}}{N/2}\right)^2\right]^{1/2}\right\}\bigg/ I_0(\pi\alpha),\quad 0\leqslant |n_{\mathrm{KB}}|\leqslant N/2 \tag{11.22}$$

其中 n_{KB} 是窗函数的抽样个数，α 是一个数字参数，可以调节它来选择最佳的旁瓣水平 / 主瓣宽度，N 是窗内抽样点的个数，且

$$I_0(x) = \sum_{k=0}^{K}\left[\frac{(x/2)^k}{k!}\right]^2 \tag{11.23}$$

是一个零阶第一类修正贝塞尔函数，K 理论上无穷大，但因为贝塞尔函数的幅度随着 k 的增加下降很快，通常用 $K = 32$ 就足够了。

11.22 式定义了点 $-N/2$ 到 $N/2-1$ 之间的窗。DFT 通常要求从 $n_{\mathrm{DFT}} = 0$ 到 $n_{\mathrm{DFT}} = N-1$，这里 n_{DFT} 是 DFT 的数据号。因此如果使用 DFT，那么凯塞 - 贝塞尔窗必须向右移动 $N/2$ 个点，满足

$$n_{\mathrm{DFT}} = n_{\mathrm{KB}} + N/2 \tag{11.24}$$

或

$$n_{\mathrm{KB}} = n_{\mathrm{DFT}} - N/2$$

则 11.22 式变成

$$w(n_{\mathrm{DFT}}) = I_0\left\{\pi\alpha\left[1.0-\left(\frac{n_{\mathrm{DFT}}-N/2}{N/2}\right)^2\right]^{1/2}\right\}\Big/ I_0(\pi\alpha),\quad 0\leqslant n_{\mathrm{DFT}}\leqslant N-1 \tag{11.25}$$

11.3.3　周期图法和周期图性质

傅里叶变换的模平方 $|F(j\omega)|^2$ 是估计的功率谱密度 $E[P(f)]$，也称为周期图。$E[P(f)]$ 可以表示为（Proakis and Manolakis, 1989; DeFatta et al., 1988）

$$E[P(f)] = \sum_{m=-\infty}^{\infty} w_{\mathrm{B}}(m)c_{xx}(m)\exp(-\mathrm{j}2\pi fm) \tag{11.26}$$

这里 $c_{xx}(m)$ 是 $x(n)$ 的延迟 m 的自相关函数，f 是频率，$w_{\mathrm{B}}(m)$ 是三角（巴特利）窗，定义为

$$w_{\mathrm{B}}(m) = 1-\frac{|m|}{N}\quad |m|\leqslant N-1 \tag{11.27}$$

比较 $x(n)$ 的真实功率谱密度 $P(f)$：

$$P(f) = \sum_{m=-\infty}^{\infty} c_{xx}(m)\exp(-\mathrm{j}2\pi fm) \tag{11.28}$$

可见由周期图给出的功率谱密度是有偏的，偏差为

$$\begin{aligned} P(f)-E[P(f)] &= \sum_{m=-\infty}^{\infty}[1-w_{\mathrm{B}}(m)]c_{xx}(m)\exp(-\mathrm{j}2\pi fm) \\ &= \frac{|m|}{N}P(f) \end{aligned} \tag{11.29}$$

对 $N\gg|m|$，偏差变得很小，则有 $E[P(f)]\to P(f)$，即周期图法是渐进无偏的。同样对于大的 N，周期图方差变成

$$\mathrm{var}[P(f)] \approx FP^2(f) \tag{11.30}$$

其中 F 取决于使用的窗函数。可见估计方差由功率谱密度的平方确定，不会随着 N 的增大而收敛到零。这表示从周期图获得的功率谱密度估计是不一致的，在连续的实现中会产生波动的 $P(f)$ 估计。

注意到通常自相关是由 N 项平均获得的，而不是选择 $N-|m|$。这两种估计都是一致和渐进无偏的，但前者的方差较小，因此比较受欢迎。

另外，当使用 DFT 以获得频谱时，相应的周期图定义为 $(1/N)|X(k)|^2$，与归一化能量有相同的尺度，尽管读者可能会注意到有些作者仍把 $(1/N)|X(k)|^2$ 当做功率谱密度。

11.3.4　修正的周期图法

韦尔奇（Welch, 1967）提议周期图法的不一致问题可以通过将一组修正周期图平均来克服。后者的每一个都包含了数据的一部分。这些部分可以是序贯的（巴特利法）或重叠的（韦尔奇法）。这种方法还能降低功率谱密度的估计方差。对巴特利法的证明请参见例 11.3。

11.3.4.1　韦尔奇法

韦尔奇法比巴特利法优越的是其功率谱密度的估计方差进一步降低。然而，它的代价是谱分辨率变得更差。在韦尔奇法中，L 个长度为 M 的数据段互相重叠，周期图是计算 L 个加窗后数据段获得的。还有，周期图用因子 U 进行归一化，补偿由于加窗过程带来的能量损失。实际上 U 等于 $1/k_2^{1/2}$，其中 k_2 是根据 11.3.2.1 节数据窗的偏差效应及对这种信号能量损失的补偿需求推导出的因子。所以

$$U = \frac{1}{M}\sum_{n=0}^{M-1} w^2(n) \tag{11.31}$$

韦尔奇的功率密度谱估计$P_{\mathrm{WE}}(f)$则是

$$P_{\mathrm{WE}}(f) = \frac{1}{L}\sum_{j=0}^{L-1} P_j(f) \tag{11.32}$$

韦尔奇估计的期望值为

$$E[P_{\mathrm{WE}}(f)] = \frac{1}{L}\sum_{j=0}^{L-1} E[P_j(f)] = E[P_j(f)] \tag{11.33}$$

它与修正周期图法的期望值相同。可以看到（Proakis and Manolakis, 1989），随着$N\to\infty$和$M\to\infty$，该值收敛到真实的功率谱密度$P(f)$。因此对于大的N和M，韦尔奇的功率谱密度估计是无偏的。在同样条件下，韦尔奇估计的方差收敛至零，即估计是一致的。韦尔奇法在无重叠的情况下（$L=K$）

$$\mathrm{var}\,[P_{\mathrm{WE}}(f)] \approx (1/K)P^2(f)$$

与巴特利法在同样条件下的方差相同。对于50%的重叠（$L=2K$），

$$\mathrm{var}\,[P_{\mathrm{WE}}(f)] \approx (9/8L)P^2(f)$$

则比巴特利法小大约9/16 = 0.56。

11.3.5 布莱克曼－图基法

在第3章建立了这个概念，即功率密度谱是由数据自相关函数的DFT得到的。也许有人会问，在周期图可以直接从数据中计算出DFT的平方的情况下，这种方法有哪些用处。首先，值得注意的是布莱克曼－图基法是1958年提出的（Blackman and Tukey, 1958），而用于快速计算DFT的FFT算法直到1965年才由Cooley和图基发表（Cooley and Tukey, 1965）。其次，布莱克曼－图基法可能包含着某些优于周期图法的地方。实际上，在下一节将会指出布莱克曼－图基法的特点是有一个大的品质因数。另外现在自相关函数也可以使用DFT的快速相关算法（参见11.3.6节）来计算。布莱克曼－图基法的过程为

(1) 计算数据的自相关函数；

(2) 在上面加一个适宜的窗；

(3) 计算加窗数据的FFT以获得功率密度谱。

与周期图法比较，我们看到平滑是通过自相关处理的平均效应得到的，而不是将几个周期图平均。

自相关函数被加窗以使其两端逐步缩减到零，因为在较大延迟处很少数据参与了（自相关函数的）计算，所以它们的估计是不精确的。逐步缩减可使这些估计的权重降低。

布莱克曼－图基法的估计$P_{\mathrm{BTE}}(f)$为

$$P_{\mathrm{BTE}}(f) = \sum_{m=-(M-1)}^{M-1} r_{xx}(m)\,w(m)\exp(-\mathrm{j}2\pi fm) \tag{11.34}$$

其中$r_{xx}(m)$是数据的自相关函数，$w(n)$是长度为$2M-1$的窗函数，当$|m|\geqslant M$时为零。

为了获得实数的估计，$w(n)$必须关于$m=0$对称，为使估计为正，它的变换也必须是正的。不是所有的窗都满足这些标准。汉宁窗和哈明窗是两个不满足的例子。

可以得到布莱克曼 - 图基估计的期望值为

$$E[P_{\mathrm{BTE}}(f)] = \sum_{m=-(M-1)}^{M-1} c_{xx}(m)w_{\mathrm{B}}(m)\exp(-\mathrm{j}2\pi fm) \tag{11.35}$$

这里 $w_{\mathrm{B}}(m)$是三角的巴特利窗。

必须满足条件 $M<N$ 以获得对频谱的附加平滑。如果 $N \gg m$ 则估计是渐进无偏的。同样，如果窗 $w(n)$的 DFT 变换 $W(k)$比真实的功率密度谱 $P(f)$窄，则

$$\mathrm{var}[P_{\mathrm{BTE}}(f)] \approx P^2(f)\left[\frac{1}{N}\sum_{m=-(M-1)}^{M-1} w^2(m)\right] \tag{11.36}$$

随着 $N/M \to \infty$，$\mathrm{var}[P_{\mathrm{BTE}}(f)] \to 0$，在这些条件下布莱克曼 - 图基估计是一致的。

11.3.6　快速相关算法

在5.3.7节指出，如果超过128个数据进行自相关，则利用相关定理（参见5.63式）并采用FFT来实现会加快计算。例如，如果$N=1024$则得到一个十倍的增速。此外，如果涉及大量的输入数据，可能会超过系统的内存容量，则可以应用重叠相加和重叠保留分块技术（5.3.8节～5.3.10节）。在布莱克曼-图基法中，当自相关函数是这样采用FFT计算出时，该方法称为频谱估计的快速相关算法。

11.3.7　功率谱密度估计方法的比较

功率谱密度估计的一种品质因数由 11.6 式给出。表 11.1 给出了四种非参数谱分析法的品质因数（Proakis and Manolakis, 1989），其中f是相关窗的 3 dB 主瓣宽度。可以看到布莱克曼 - 图基法的品质是最好的。除了周期图法以外，随着N的增加频率分辨率同时增加（f降低），品质得以保持。

表 11.1　功率谱密度估计的品质因数 Q

估计方法	条件	Q	注释
周期图	$N\to\infty$	1	不一致，与 N 无关
巴特利	$N, M\to\infty$	$1.11Nf$	品质随着数据长度的增加而提高
韦尔奇	$N, M\to\infty$，50% 重叠	$1.39Nf$	品质随着数据长度的增加而提高
布莱克曼 - 图基	$N, M\to\infty$，三角窗	$2.34Nf$	品质随着数据长度的增加而提高

需要非常小心以及一些试验计算才能保证满意的结果。从折中的观点上来看，布莱克曼-图基法是临界最优的，但从其他的考虑角度出发会偏好别的方法。

11.4　现代参数估计法

本章前面介绍的利用周期图和 FFT 的非参数法，如前所述其目标限制于对较短记录的较低的频谱分辨，且需要加窗以防止频谱泄漏。这些困难可能会通过参数法加以克服（Burg, 1968; Nuttall, 1976; Ulrych and Clayton, 1976; Marple, 1980; Cadzow, 1979, 1982; Graupe et al., 1975; Kay, 1980; Friedlander, 1982）。需要付出的代价是广泛研究每种过程的适用模型，确定所选模型的阶数以满足数据的表达（Whittle, 1965; Jenkins and Watts, 1968; Box and Jenkins, 1976; Chatfield, 1979; Akaike, 1969; 1973, 1974, 1978, 1979; Shibata, 1976; Rissanen, 1983），以及计算模型参数（Proakis and Manolakis, 1989; Makhoul, 1975; Levinson, 1947; Durbin, 1959; Priestley, 1981; Wold, 1954; Chatfield, 1984）。获得的收益是频谱分辨率提高，适用于短长度数据，以及避免了频谱泄漏、扇形损失、频谱混淆及窗的偏差效应。因为这些参数法的重要性，我们将介绍最常使用的自回归模型。不过，尽

管具有比非参数法提高的性能，这些参数法确实存在一些缺点，可能采用替代的现代方法如顺序或自适应（Friedlander, 1982; Kalouptsidis and Theodoridis, 1987）和最大似然法（Capon, 1969; Lacoss, 1971）来避免。

总地来说，参数法要求数据的参数模型，属于时间序列分析（Jenkins and Watts, 1968; Box and Jenkins, 1976; Priestley, 1981）的一个确定分支，结合了将数据描述为白噪声激励一个线性系统后的输出。这个系统被表示为一个使用模型参数的多项式传递函数。数据的谱就根据传递函数来计算。

11.5 自回归频谱估计

在这个模型中，数字化信号被模型化为一个自回归（AR）时间序列加上一个白色噪声的误差项。频谱则可以从AR模型的参数和误差项的方差中获得。模型参数通过求解一组线性方程来获得，即最小化所有数据的均方误差项（白色噪声的功率）。有许多方法来求解方程，将在下面的内容中详细介绍。一个重要的考虑是选择AR模型中的项数，或者说它的阶数。如果阶数太低，则功率密度谱估计会过于平滑，那么有些谱峰就模糊了。如果阶数过高，就会带来伪峰。因此，确定每组数据的最佳模型阶数是非常重要的，我们将讨论这个问题。这种模型适用于功率密度谱具有尖锐谱峰的信号。其他模型如移动平均或自回归移动平均模型适用于不满足该条件的信号。由于自回归方法的方程是可解的，如果可能的话就使用它。参考文献给出了更为详细的资料（Pardey, Roberts and Tarassenko, 1996; Kay, 1988; Candy, 1989; Proakis and Manolakis, 1989; Marple, 1987; Clarkson, 1993）。

11.5.1 自回归模型和滤波器

在一个时间序列的AR模型中，序列的当前值$x(n)$被表示为过去值的线性函数加上一个误差项$e(n)$，即

$$\begin{aligned} x(n) = &-a(1)x(n-1) - a(2)x(n-2) - \ldots - a(k)x(n-k) - \ldots \\ &- a(p)x(n-p) + e(n) \end{aligned} \tag{11.37}$$

这个方程包含了p个过去项，或者说是个p阶模型。可以将其更紧凑地写为

$$x(n) = -\sum_{k=1}^{p} a(k)x(n-k) + e(n) = -\sum_{k=1}^{p} a(k)z^{-k}x(n) + e(n) \tag{11.38}$$

其中z^{-k}是后向移动算子，表示延迟k个抽样间隔。重写11.38式：

$$x(n) + \sum_{k=1}^{p} a(k)z^{-k}x(n) = \left(1 + \sum_{k=1}^{p} a(k)z^{-k}\right)x(n) = e(n) \tag{11.39}$$

因此

$$x(n) = \frac{e(n)}{1 + \sum_{k=1}^{p} a(k)z^{-k}} \tag{11.40}$$

将比率项$x(n)/e(n)$提出得到

$$\frac{x(n)}{e(n)} = \frac{1}{1 + \sum_{k=1}^{p} a(k)z^{-k}} = H(z) \tag{11.41}$$

这里$H(z)$被解释为一个全极点IIR数字滤波器的z变换，滤波器系数为$a(k)$。该滤波器称为自回归（AR）滤波器，如图11.15所示。根据11.41式，$x(n)$可以看成是这个滤波器由随机输入$e(n)$产生的输出。$e(n)$代表根据模型进行预测的值$\hat{x}(n)$与真实的数据值$x(n)$间的误差。$e(n)$经常假设为具有白色噪声的性质，也就是具有高斯概率密度分布和一个均匀的功率密度谱。这样$x(n)$可以看成是一个白色噪声源激励AR滤波器产生的。滤波器的频响$H(f)$通过替换11.39式（参见4.4节）中的$z = \mathrm{e}^{\mathrm{j}\omega T}$而获得，其中$\omega$是角频率，$T$是抽样周期。所以

$$H(f) = \frac{1}{1 + \sum_{k=1}^{p} a(k)\mathrm{e}^{-\mathrm{j}k\omega T}} \tag{11.42}$$

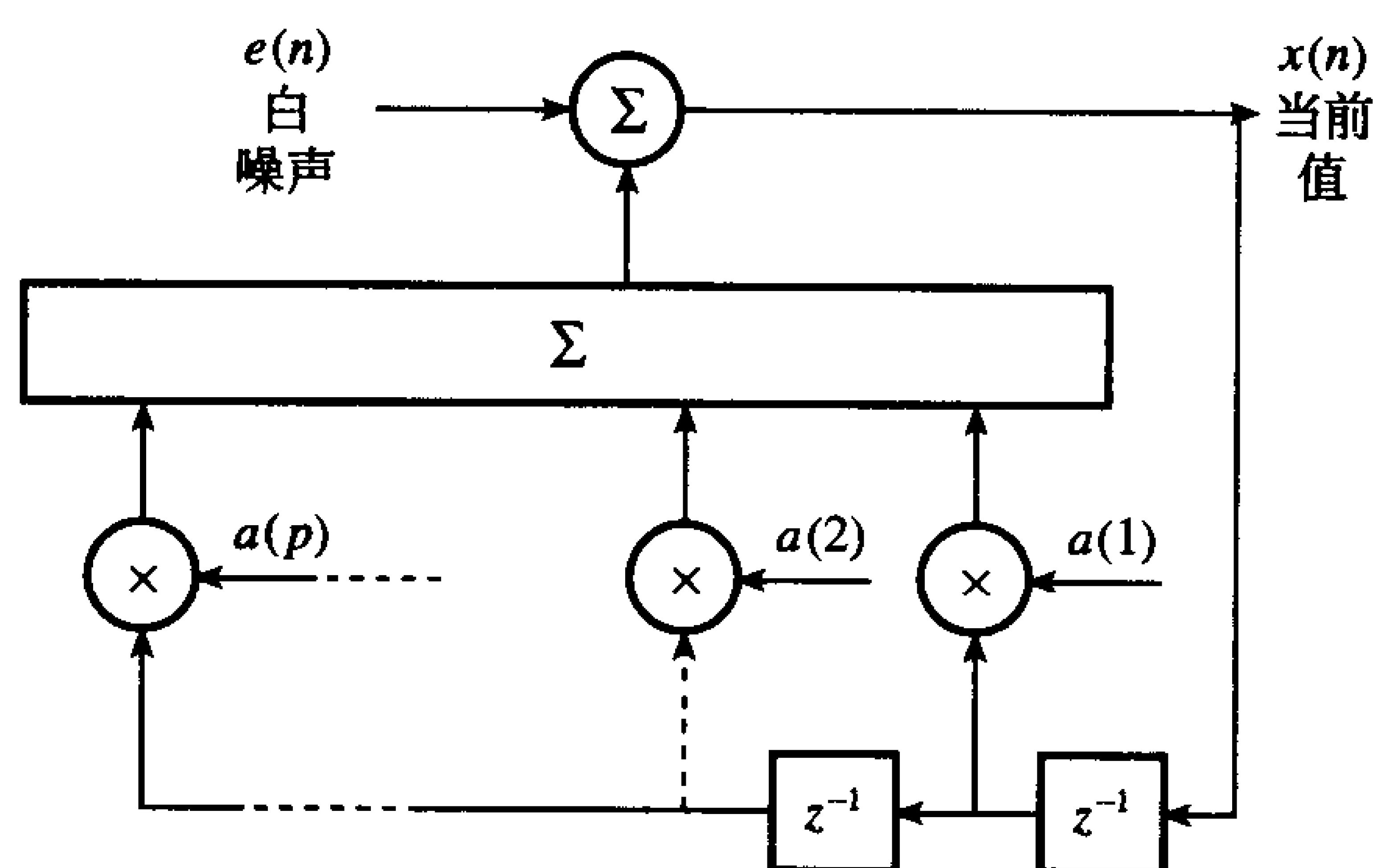


图 11.15　AR 滤波器

11.5.2　AR 序列的功率谱密度

需要了解AR序列$x(n)$的功率谱密度$P_x(f)$。这关系到白色噪声的误差信号$P_e(f)$的功率谱密度，即方差$\sigma_e^2(n)$，则

$$P_x(f) = |H(f)|^2 P_e(f) = |H(f)|^2 \sigma_e^2(n) = \frac{\sigma_e^2(n)}{\left|1 + \sum_{k=1}^{p} a(k)\mathrm{e}^{-\mathrm{j}k\omega T}\right|^2} \tag{11.43}$$

白色噪声的方差是其均方值，即$e(n)$的均方值，即后面提到的E。可以确定模型参数，$\sigma_e^2(n)$（或E）就可以通过这些参数而确定，由此得到了功率密度谱。

11.5.3　模型参数的计算——Yule-Walker 方程

最优模型参数是那些能够对每个抽样点$x(n)$都最小化误差项$e(n)$的量，表示为如11.38式的方程。这些误差可通过重排11.38式得到：

$$e(n) = x(n) + \sum_{k=1}^{p} a(k)x(n-k) \tag{11.44}$$

需要一种对于所有抽样$x(n)$（$1 \leqslant n \leqslant N$）的总误差的测度。每个误差$e(n)$可能是正的或负的，所以对大量的抽样点来说平均误差趋于很小。因此平均误差不适于作为模型精确与否的测度，而是使用均方误差的量，因为所有的均方误差项都是正的。均方误差表示为

$$E = \frac{1}{N}\sum_{n=1}^{N} e^2(n) = \frac{1}{N}\sum_{n=1}^{N}\left(x(n) + \sum_{k=1}^{p} a(k)x(n-k)\right)^2 \tag{11.45}$$

可以看到 $E=\sigma_e^2(n)$。现在需要的是选择模型参数来最小化E。每个参数的最优值是通过设置11.45式对模型参数的偏导为零而得到的。由此对第k个参数，有

$$\frac{\partial E}{\partial a(k)}=\frac{2}{N}\sum_{n=1}^{N}\left(x(n)+\sum_{k=1}^{p}a(k)x(n-k)\right)\frac{\partial}{\partial a(k)}\sum_{k=1}^{p}a(k)x(n-k)=0,1\leqslant k\leqslant p \tag{11.46}$$

现在

$$\frac{\partial}{\partial a(k)}\sum_{k=1}^{p}a(k)x(n-k)=x(n-k)$$

所以11.46式简化为

$$\frac{\partial E}{\partial a(k)}=\frac{2}{N}\sum_{n=1}^{N}\left(x(n)+\sum_{k=1}^{p}a(k)x(n-k)\right)x(n-k)=0 \tag{11.47}$$

对第k个参数：

$$\frac{1}{N}\sum_{n=1}^{N}\left(\sum_{k=1}^{p}a(k)x(n-k)\right)x(n-k)=-\frac{1}{N}\sum_{n=1}^{N}x(n)x(n-k) \tag{11.48}$$

将11.48式的左端写出，例如对$k=1$的情况有

$$\begin{aligned}&\frac{1}{N}\sum_{n=1}^{N}(a(1)x(n-1)+a(2)x(n-2)+\ldots+a(p)x(n-p))x(n-1)\\&=\frac{1}{N}(a(1)x(0)x(0)+a(2)x(-1)x(0)+\ldots+a(p)x(1-p)x(0))\\&+\frac{1}{N}(a(1)x(1)x(1)+a(2)x(0)x(1)+\ldots+a(p)x(2-p)x(1))+\ldots\\&+\frac{1}{N}(a(1)x(N-1)x(N-1)+a(2)x(N-2)x(N-1)+\ldots+a(p)x(N-p)x(N-1))\end{aligned}$$

检查每一行的第一项，可以看出它们加起来构成了序列$x(n)$的零延迟自相关函数$R_{xx}(0)$乘上$a(1)$。同样，第二项加起来构成了一延迟的自相关函数$R_{xx}(-1)$乘上$a(2)$，第p项加起来等于$R_{xx}(-(p-1))$乘$a(p)$。由于自相关函数$R_{xx}(-j)=R_{xx}(j)$，表达式可以写为

$$R_{xx}(0)a(1)+R_{xx}(1)a(2)+\ldots+R_{xx}(k-1)a(k)+\ldots+R_{xx}(p-1)a(p)$$

11.48式的右端等于$-R_{xx}(1)$。联立左端和右端得到

$$R_{xx}(0)a(1)+R_{xx}(1)a(2)+\ldots+R_{xx}(k-1)a(k)+\ldots+R_{xx}(p-1)a(p)=-R_{xx}(1) \tag{11.49}$$

对k的每个取值，$1\leqslant k\leqslant p$，都可写出一个类似方程。归总这些方程可写成矩阵的形式：

$$\begin{pmatrix}R_{xx}(0)&R_{xx}(1)&\cdots&R_{xx}(p-1)\\R_{xx}(1)&R_{xx}(0)&\cdots&R_{xx}(p-2)\\\vdots&\vdots&\vdots&\vdots\\R_{xx}(p-1)&R_{xx}(p-2)&\cdots&R_{xx}(0)\end{pmatrix}\begin{pmatrix}a(1)\\a(2)\\\vdots\\a(p)\end{pmatrix}=-\begin{pmatrix}R_{xx}(1)\\R_{xx}(2)\\\vdots\\R_{xx}(p)\end{pmatrix} \tag{11.50}$$

模型参数$a(k)$现在可以从这组称为Yule-Walker（YW）的方程中获得。用矩阵符号表示11.50式有

$$\mathbf{R}_{xx}(k-j)\mathbf{a}(k)=-\mathbf{R}_{xx}(k) \tag{11.51}$$

因此原则上，

$$\mathbf{a}(k) = -\mathbf{R}_{xx}^{\mathrm{T}}(k-j)\mathbf{R}_{xx}(k) \tag{11.52}$$

可以看出 $\mathbf{R}_{xx}(k-j)$是对称的，由于主对角线上的每个元素都是相同的（等于 $R_{xx}(0)$），因此称为是 Toeplitz 的。只要 $x(n)$不是纯的正弦量，矩阵还是正定的。有数种方法求解 11.50 式以得到 $a(k)$。

11.45 式允许计算 E，但可以找到另一种使用自相关函数和 $a(k)$的表达式，如下所示。假定 $a(k)$和 $x(n)$是实的，扩展 11.45 式得到

$$\begin{aligned} E &= \frac{1}{N}\sum_{n=1}^{N}\left\{x(n)+\sum_{k=1}^{p}a(k)x(n-k)\right\}\left\{x(n)+\sum_{k=1}^{p}a(k)x(n-k)\right\} \\ &= \frac{1}{N}\sum_{n=1}^{N}\left[\left\{x(n)+\sum_{k=1}^{p}a(k)x(n-k)\right\}x(n)\right. \\ &\quad \left.+\left\{x(n)+\sum_{k=1}^{p}a(k)x(n-k)\right\}\sum_{k=1}^{p}a(k)x(n-k)\right] \end{aligned} \tag{11.53}$$

利用 11.47 式，对所有的 k 都适用，在 11.53 式中有

$$\frac{1}{N}\sum_{n=1}^{N}\left\{x(n)+\sum_{k=1}^{p}a(k)x(n-k)\right\}\sum_{k=1}^{p}a(k)x(n-k) = 0$$

因此 11.53 式简化为

$$\begin{aligned} E &= \frac{1}{N}\sum_{n=1}^{N}\left\{x(n)+\sum_{k=1}^{p}a(k)x(n-k)\right\}x(n) \\ &= \frac{1}{N}\sum_{n=1}^{N}x^2(n)+\frac{1}{N}\sum_{n=1}^{N}\left(\sum_{k=1}^{p}a(k)x(n-k)\right)x(n) \\ &= R_{xx}(0)+\sum_{k=1}^{p}\frac{1}{N}\sum_{n=1}^{N}a(k)x(n)x(n-k) \end{aligned}$$

所以最后有

$$E = R_{xx}(0)+\sum_{k=1}^{p}a(k)R_{xx}(k) \tag{11.54}$$

11.54 式或 11.45 式和从 11.52 式获得的模型参数现在可以插入到 11.43 式，从而获得自回归功率密度谱。然而，从方程 11.50 中求解 $a(k)$的可行方法和模型阶数 p 的选择，是必须首先考虑的。

11.5.4　Yule-Walker 方程的求解

由 11.45 式给出的均方误差 E，其计算使用的是现存的抽样数据 $x(n)$，从 $n=1$ 到 $n=N$。前面的和后续的 $x(n)$值被预置为零。如前面所说，这等效于对数据加窗，对于非参数法频谱估计的情况，它导致旁瓣产生频谱混淆，降低了分辨率。然而，这并不适用于自回归法。可以看到（Kay, 1988）这种方法隐含了对延迟超过 p 的自相关函数的估计，而实际上 $x(n)$里并没有相应的数据。因此自回归法提供了更高的谱分辨率。然而，可能通过如在现有数据上的偏置算法来提高谱估计精度。在该方法中，不需要 $0 \leqslant n < N$ 的 n 值，所以不用将它们置零。这些方法是本节的主题。一个更加完整的处理在如 Kay(1988)和 Candy(1989)中有专门介绍。算法的程序软件包在 Kay(1988)和其他文献中给出。

11.5.4.1 自相关算法

自相关算法是基于 11.45 式的均方根误差表达。Levinson-Durbin 算法（Kay, 1988; Pardey, Roberts and Tarassenko，1996）提供了一种高效计算 11.50 式的 YW 方程，从而得到模型参数的算法。该算法的频率分辨率比其他算法低，因此不适用于较短的数据记录。

11.5.4.2 协方差方法

在这种方法中，11.45式求和的限可以修正为$n=p$到$n=N$，这意味着对于自相关函数的计算，只要求$x(n)$的有效值。此外，平均是对$N-p$项做的而不是N项。因此，11.45 式变成

$$E=\frac{1}{N-p}\sum_{n=p}^{N}\left(x(n)+\sum_{k=1}^{p}a(k)x(n-k)\right)^2 \tag{11.55}$$

11.50 式等效于

$$\begin{pmatrix} C_{xx}(1,1) & C_{xx}(1,2) & \cdots & C_{xx}(1,p) \\ C_{xx}(2,1) & C_{xx}(2,2) & \cdots & C_{xx}(2,p) \\ \vdots & \vdots & \vdots & \vdots \\ C_{xx}(p,1) & C_{xx}(p,2) & \cdots & C_{xx}(p,p) \end{pmatrix}\begin{pmatrix} a(1) \\ a(2) \\ \vdots \\ a(p) \end{pmatrix}=-\begin{pmatrix} C_{xx}(1,0) \\ C_{xx}(2,0) \\ \vdots \\ C_{xx}(p,0) \end{pmatrix} \tag{11.56}$$

其中

$$C_{xx}(j,k)=\frac{1}{N-p}\sum_{n=p}^{N}x(n-j)x(n-k) \tag{11.57}$$

E由下式给出

$$E=C_{xx}(0,0)+\sum_{k=1}^{p}a(k)C_{xx}(0,k) \tag{11.58}$$

$p\times p$矩阵$C_{xx}(j,k)$是共轭和半正定的。11.56 式可以使用 Cholensky 分解法（Lawson and Hanson, 1974）来求解。只有$N-p$个延迟分量需要累加，所以对于长度较短的数据，会有一些尾端效应。协方差法的谱分辨率结果要比自相关法好。

11.5.4.3 修正协方差法

在修正协方差法中前向和后向预测误差估计的均值被最小化（Kay, 1988; Candy, 1989）。11.56式和 11.58 式仍旧使用，但 11.57 式做以下修改：

$$C_{xx}(j,k)=\frac{1}{2(N-p)}\left\{\sum_{n=p}^{N}x(n-j)x(n-k)+\sum_{n=1}^{N-p}x(n+j)x(n+k)\right\} \tag{11.59}$$

$p\times p$矩阵$C_{xx}(j,k)$仍是共轭和半正定，11.56 式可以使用 Cholensky 分解法（Lawson and Hanson, 1974）来求解。该方法不保证一个稳定的全极点滤波器，但常常会得到它。它产生一个统计上稳定的高分辨率频谱估计。

11.5.4.4 Burg 算法

这种算法超出了目前的研究范围。它能从 AR 数据中产生精确的频谱估计。

11.5.5 模型阶数

对于每组实测数据，必须小心选择能够符合数据的自回归模型的最佳阶数，因为它依赖于数据的统计特性。举一个EEG数据中的例子，其中每个数据段需要不同的模型阶数（Pardey, Roberts and Tarassenko, 1996）。一般希望较低的模型阶数，这样只需要拟合较少的参数。然而，如果阶数过低，频谱估计会太平滑。另一方面，过高的阶数导致伪峰和谱估计的不稳定。两个最常用的阶数估计参数由 Akaike 提出。它们是最终估计误差$FPE(p)$（Akaike, 1969）

$$FPE(p)=\frac{N+p}{N-p}E(p) \tag{11.60}$$

和 Akaike 信息判据 $AIC(p)$（Akaike, 1974）

$$AIC(p)=N\ln E(p)+2p \tag{11.61}$$

它被推荐用于短数据记录，而 $FPE(p)$ 则推荐用于较长的数据记录。一个实际的用法是努力选择 p 使 $FPE(p)$ 和 $AIC(p)$ 最小。

11.6　估计方法的比较

在非参数法中，布莱克曼-图基法具有大的品质因数，因此较受青睐，尽管为方便起见也可能使用其他方法。

参数法提供了更高的频率分辨率，且避免了加窗函数的效应。Capon 法、最大似然法得到具有最小方差的无偏估计，其频谱分辨率介于 Burg 法或无约束最小二乘法和非参数法之间。自适应滤波法更加重视最近的数据，因此适用于非平稳数据。

11.7　应用举例

11.7.1　利用基于 DFT 的频谱分析来辨别脑部疾病

从 DFT 变换得到的幅度和相位谱被应用于辨别脑部疾病过程，如亨廷顿（Huntington）症、精神分裂症、帕金森（Parkinson）症和正常状态，这是通过分析所选择测试者的脑电波（EEG）中偶发性负波（CNV）频谱中的谐波分量（Jervis et al., 1993）而确定的。

CNV 是一个事件关联电压（ERP），表现为头皮部位上一个负电压，可以由一个适当的听觉刺激来激发。CNV 波形示意图见图 11.16。CNV 是介于听觉刺激加入点 S_1 和 S_2 之间的负波形。基于某些理由相信人们在遭受上述病症之一时会影响 CNV 波形。通过确定波形中适当部分的频谱，并加以统计处理，也许可以区分不同的病症类型。

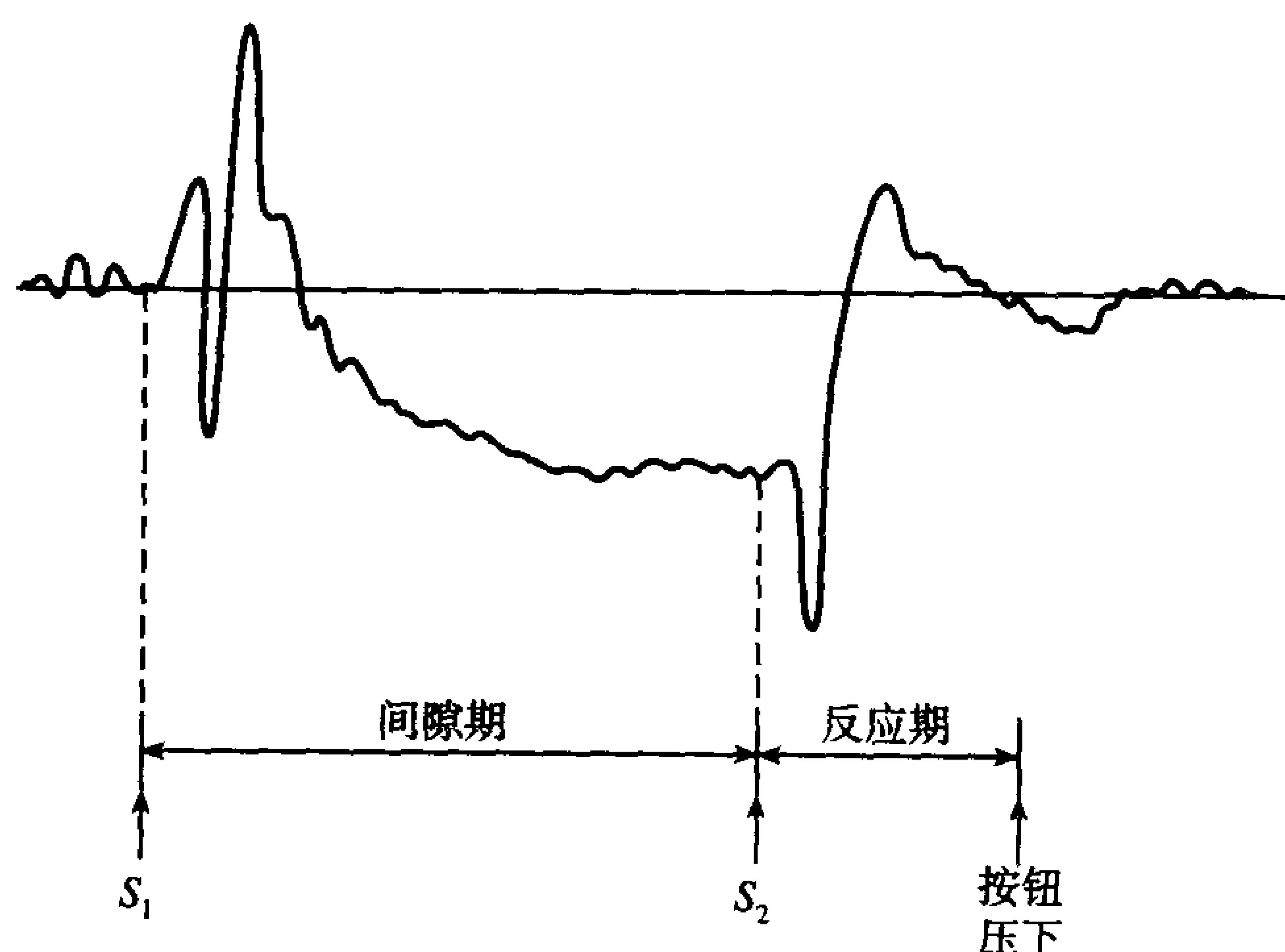


图 11.16　CNV 示意波形图

从各种病症的测试者中，利用专用设计的信号处理装置系统获得不同的 CNV（Jervis and Saatchi，1990；Saatchi and Jervis，1991）。数据经预处理以降低背景的 EEG 影响和 CNV 波形中的视觉运动。信号的均值水平被消除，即可进行不同时间段的比较，保证视觉运动消除算法工作正常。

均值水平消除造成刺激前后基线的正向移动。因此通过减去相应的不同部分各自的均值就能校正基线。通过数字低通滤波以滤除 EEG 中不需要的高频分量，经常使用 FIR 滤波器而不用 IIR 滤波器，因其会造成波形的失真。再应用比例相减（Jervis et al., 1989b）的视觉运动消除算法以去除视觉运动。将各目标类别的 8 个预处理 CNV 波形的平均显示在图 11.17 ~ 图 11.20，目标类别分别是正常人、亨廷顿症、精神分裂症、帕金森症。

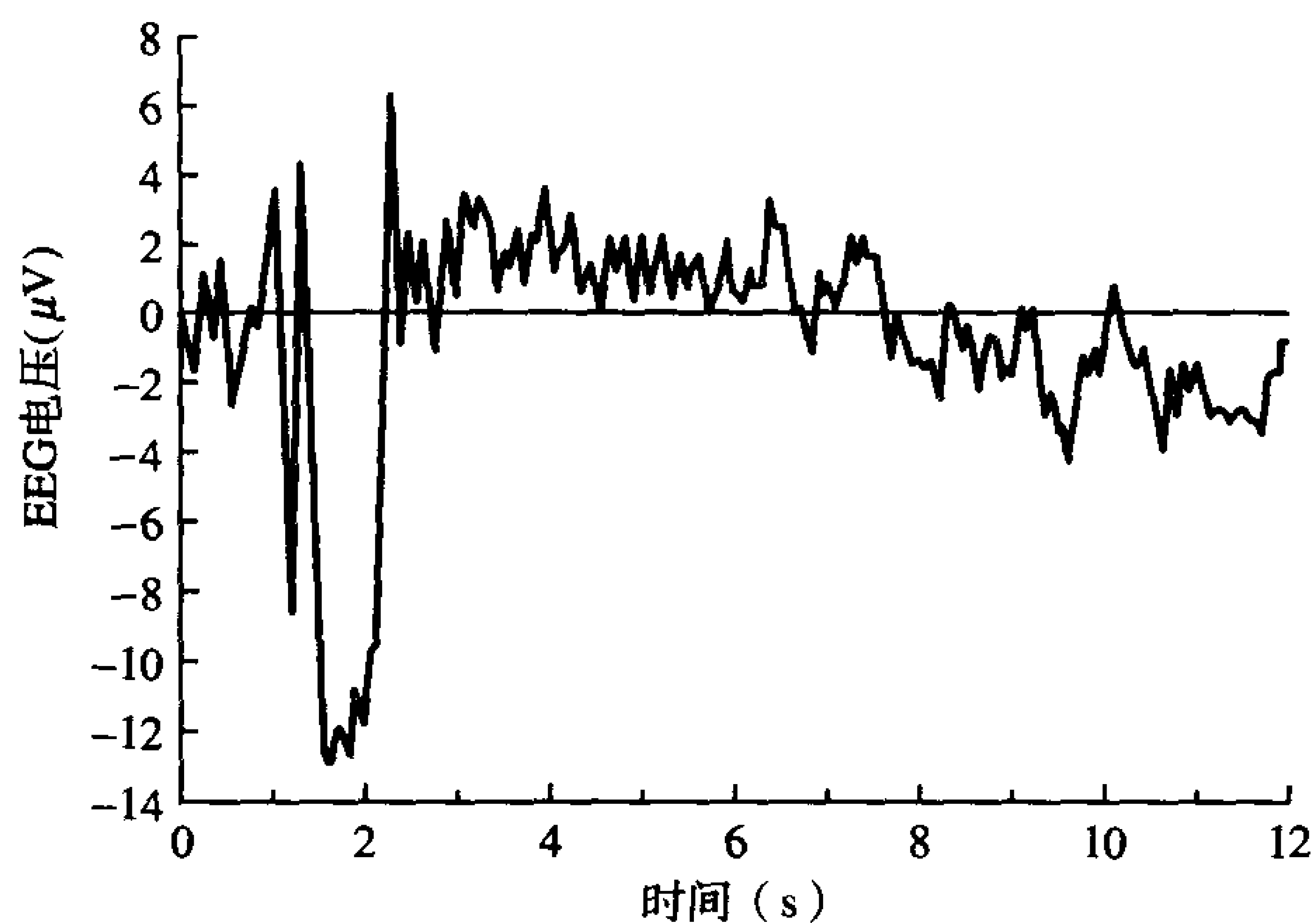


图 11.17　正常人的预处理和平均 CNV 波形

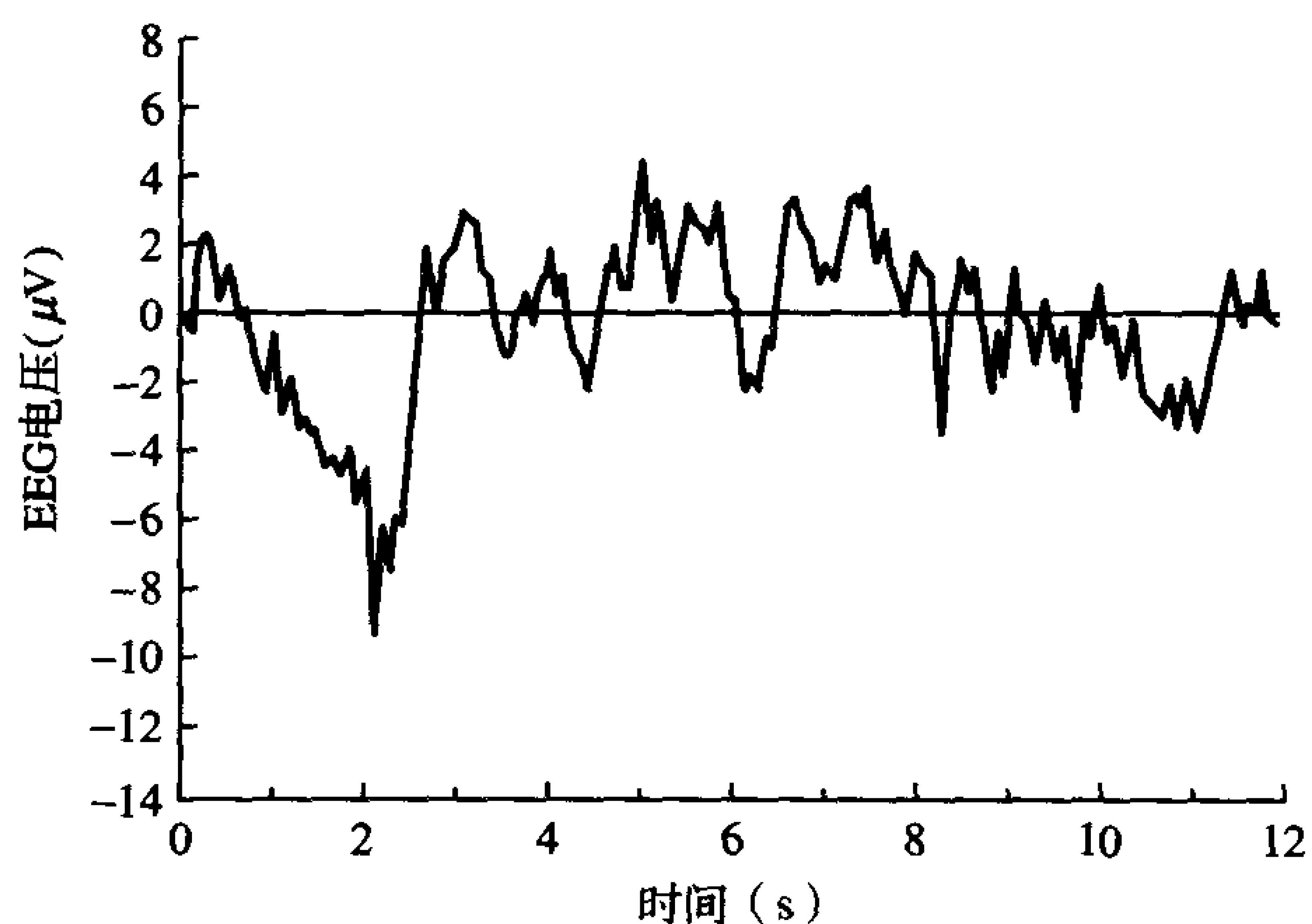


图 11.18　亨廷顿症患者的预处理和平均 CNV 波形

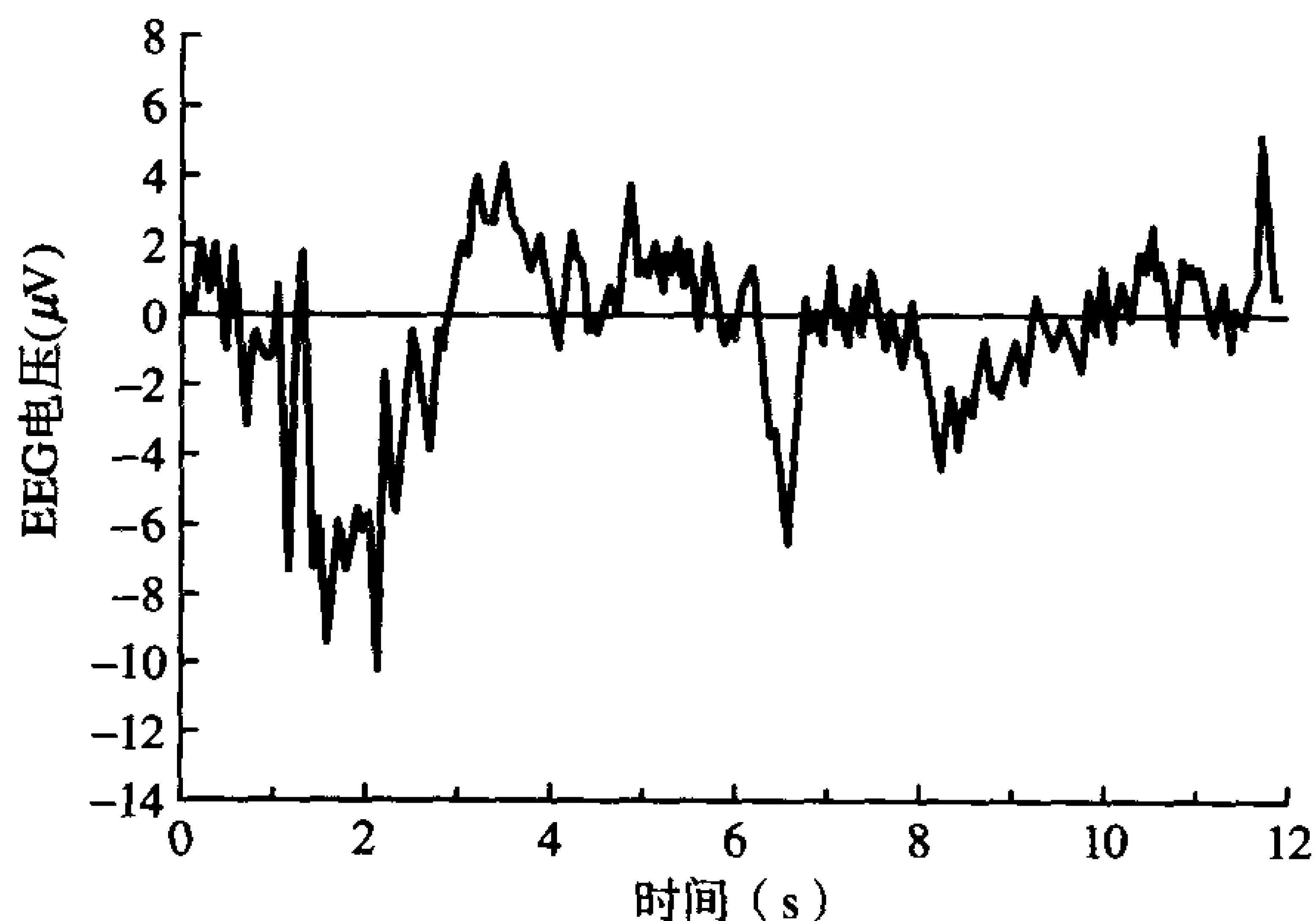


图 11.19　精神分裂症患者的预处理和平均 CNV 波形

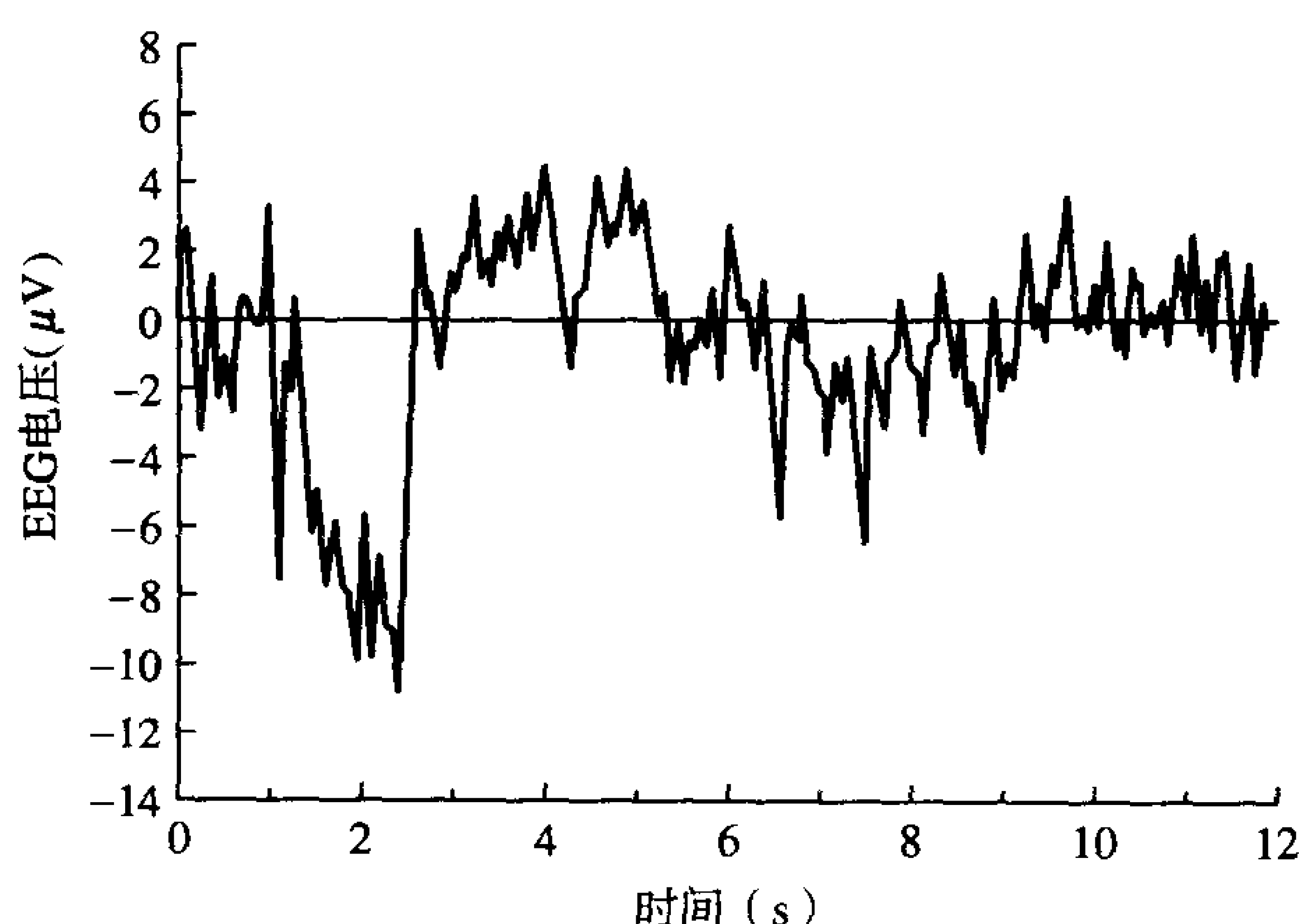


图 11.20　帕金森症患者的预处理和平均 CNV 波形

每个实验者的两个 512 ms（64 个样点）CNV 波形段将加上凯塞 - 贝塞尔窗。实验指出窗口参数值α为0.75时，窗的旁瓣水平和主瓣宽度之间能够达到较好的平衡。960个增补零值被加在64个样点数据上以降低扇形损失。计算这组1024个数据的DFT。对所产生频谱的前96个谐波分量应用四种统计检验。这些检验请参见 Jervis et al.(1983)。将检验命名为最近和最远幅度均值检验、刺激前后的幅度差别检验、圆周方差的瑞利检验和圆周方差的修正瑞利检验。这些检验产生了大量的检验统计量。

为了降低检验统计量的数目，选择其中最具有识别力的，它们均属于单变量检验、t检验和逐次差别分析（stepwise discriminant analysis）。这个过程由 Jervis et al.(1993)给出，使用统计程序包 SAS（SAS, 1982）来实现。

目前，个体的区分采用差别分析（Morrison, 1976）。同样，详细情况在 Jervis et al.(1993)中给出，SAS 包用于实现。结果总结于表 11.2。

表 11.2　脑部疾病的区分结果总结

目标类型		试验的鉴别成功率(%)	
类型 1	类型 2	类型 1	类型 2
HD	可控	100	100
精神分裂症	可控	95	100
PD	可控	93.8	87.5
HD	精神分裂症	100	90.9
HD	PD	90.9	81.8
精神分裂症	PD	81.3	93.8

HD，亨廷顿症患者；PD，帕金森症患者；可控，年龄和性别匹配的试验者。

这些结果显示：在CNV波形中结合统计技术应用频谱分析，可以在很高精度上成功区分亨廷顿症、精神分裂症、帕金森症患者和正常人。

11.7.2　应用自回归模型的 EEG 频谱分析

对一个事件关联电压（ERP）的信号记录仿真，是将一段测量的脑电图（EEG）信号和仿真的 ERP 相加得到的。ERP 的记录仿真和 ERP 仿真在图 11.21(a)中给出。图 11.21(b)显示了 ERP 记录仿真的功率密度谱，它使用了 FFT 变换的方法，再应用 11.5 节给出的自回归确定频谱法得到其信号频谱。11.5.4.2 节的协方差法用于计算模型参数。ERP 记录仿真的 AR 谱在图 11.21(c)中给出，自回归模型的阶数为 50，这比从 FFT 得到的更平滑。测量的 EEG 的 AR 谱也在图 11.21(d)给出，

可以加以比较。就像所预料的那样，它在2 Hz以下所包含能量比仿真的ERP要小，因为ERP是集中于低频上的。ERP的AR谱请参见图11.21(e)，可以看到其信号能量确是限于2 Hz带宽内，证实了前面的观察。比较图11.2(d)和图11.2(e)，发现低于2 Hz的能量主要是与ERP有关而与EEG无关。最后图11.21(f)也给出了ERP的AR谱，但其模型阶数取为6而不是50。这些谱是同样的，说明一个较低的模型阶数对确定窄带ERP的谱就足够了，同时需要一个更高阶的模型来确定EEG的谱，因为它具有更宽的频带。

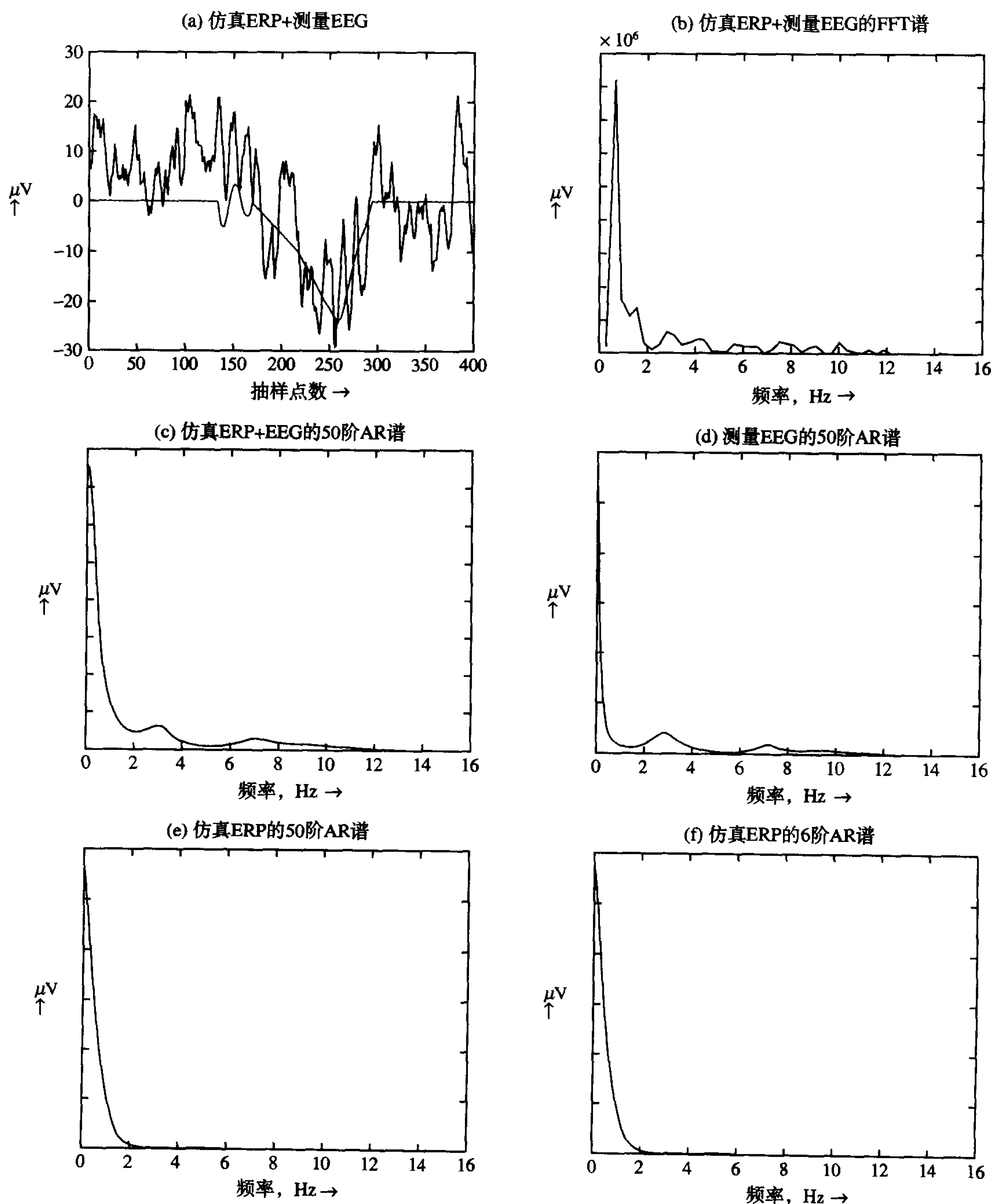


图11.21　一个自回归谱的图例

11.8　小结

本章曾指出频谱估计的参数法会产生更为可靠的结果。由于它们可以自动操作，可能比基于周期图的非参数法更受欢迎。后者的可靠性较差，它们需要专门的操作技术来保证获得有意义的结果。然而，在区分脑部疾病的临床应用中，发现它们具有很好的作用。

11.9　处理过的实例

例 11.3　说明功率谱密度的巴特利估计是渐进无偏的，且随着数据段的个数增加，估计方差随之递减，还有谱估计是一致的。修正周期图法对频率分辨率有何影响？

解：

首先计算 K 个 M 点数据的非重叠段的周期图。若总数据点为 N，则 $K=N/M$。K 个周期图的平均就得到了巴特利功率密度谱估计 $P_{BE}(f)$，且

$$P_{BE}(f)=\frac{1}{K}\sum_{j=0}^{K-1}P_j(f) \tag{11.62}$$

j 表示数据段的序号，$P_j(f)$ 是相应的第 j 个功率谱密度。

检查 $P_{BE}(f)$ 的期望值得到

$$\begin{aligned}E[P_{BE}(f)]&=\frac{1}{K}\sum_{j=0}^{K-1}E[P_j(f)]\\&=E[P_j(f)]\\&=\sum_{m=-(M-1)}^{M-1}\left(1-\frac{|m|}{M}\right)c_{xx}(m)\exp(-\mathrm{j}2\pi fm)\end{aligned} \tag{11.63}$$

对于 $M\gg|m|$，巴特利窗项 $1-|m|/M$ 趋近于单位 1，$E[P_{BE}(f)]$ 变成真实的功率谱密度 $P(f)$。因此 $P(f)$ 的巴特利估计是渐进无偏的。

现在转向方差，

$$\mathrm{var}[P_{BE}(f)]=\frac{1}{K^2}\sum_{j=0}^{K-1}\mathrm{var}[P_j(f)]=\frac{1}{K}\mathrm{var}[P_j(f)] \tag{11.64}$$

所以方差的大小与数据段的个数 K 成反比，共有 N 个数据被分段。因为方差随着 K 的增大而递减，$P(f)$ 的巴特利估计是一致的。然而，由于参与计算周期图的数据个数按照因子 K 减小到 $M=N/K$，谱分辨率也降低同样的因子。所以从巴特利法获得的主瓣宽度是从全部数据 N 获得主瓣宽度的 K 倍。

习题

11.1　(1) 一个波形按 30 kHz 抽样。对前 524 288 个样点应用快速傅里叶变换。计算第一个谐波的频率和谱的频率分辨率。

(2) 如果波形的真实频谱包含了一个 5.7505 Hz 的正弦分量，你怎样修正数据以确保该分量被谱估计正确地表达？

11.2　一个波形以 8 kHz 抽样，抽样值分别是 1.0，1.0，1.0，1.0，1.0，1.0，1.0，1.0 V。这些数据再通过一个函数窗，相应的抽样值变成 0，0.5，1.0，1.0，1.0，1.0，0.5，0.0。确定加窗数据的 DFT。

11.3 计算习题 11.2 数据的扇形损失和等效噪声带宽。

11.4 参考图 11.10、图 11.12 和图 11.13，估计下列窗的等效噪声带宽和处理增益，每种窗使用 8 个抽样值。

(1) 矩形窗；

(2) 图基（余弦缩减）窗，$\alpha = 0.1$；

(3) 图基窗，$\alpha = 0.5$；

(4) 哈明窗，$\alpha = 0.54$；

(5) 凯塞 - 贝塞尔窗，$\alpha = 2.0$；

(6) 凯塞 - 贝塞尔窗，$\alpha = 4.0$。

11.5 一个远程通信的脉冲波形每隔 0.167 μs 被抽样，得到抽样值 0，0，1，1，1，1，1，0 V。其非零值通过一个 $\alpha = 4.0$ 的凯塞 - 贝塞尔窗。使用一个 8 点 FFT（或 DFT）计算加窗后的脉冲能量谱。（从图 11.13(c)获得窗的估计数据。）

11.6 确定习题 11.5 中窗的扇形损失、处理损失和最坏处理损失。

11.7 对于习题 11.5 的数据，确定 $\alpha = 0.54$ 的哈明窗、$\alpha = 4.0$ 的凯塞 - 贝塞尔窗、$\alpha = 0.5$ 的图基窗和矩形窗的最坏处理损失以及第一旁瓣幅度。将结果制表，并选择最适宜的窗。（使用图 11.13(c)、图 11.12(a)和图 11.10(b)来获得窗函数的估计。）

11.8 从一个按 8 kHz 抽样的波形中得到抽样电压 0，4.0，2，4，1.0，−1.0，−3.8，−1.3，0 V。计算和画出能量谱。

(1) 对数据应用 $\alpha = 2.0$ 的凯塞 - 贝塞尔窗；

(2) 根据 11.21 式修改数据。

$$S(n) = w(n)\left[s(n) - \frac{\sum_{n=0}^{N-1} w(n)s(n)}{\sum_{n=0}^{N-1} w(n)}\right] \times \left[\frac{N}{\sum_{n=0}^{N-1} w^2(n)}\right]^{1/2}$$

其中 $w(n)$是 $\alpha = 2.0$ 的凯塞 - 贝塞尔窗的抽样值；

(3) 解释从(1)、(2)所获得的结果不同的原因。

11.9 从抽样数据序列{0, 1, 0, 1, 0, 1, 0, 1}获得能量谱，将它与你所期望的方波的频谱做比较。

11.10 应用巴特利修正周期图法获得习题 11.9 的数据的能量谱，将其分成两个不重叠的数据段，分别为{0, 1, 0, 1}和{0, 1, 0, 1}。

11.11 应用韦尔奇修正周期图法获得习题 11.9 的数据的能量谱，将其分成三个相等长度的数据段，重叠率 50%。

11.12 现在假定习题 11.9 的数据包含了一个随机噪声分量，继而新的数据序列变成{0.763, 1.656, 0.424, 1.939, 0.133, 1.881, 0.328, 1.348}。计算该加噪数据的能量谱，并与原始数据的相比较。从抽样数据估计其信噪比。

11.13 现在对习题11.12的数据重复巴特利修正周期图法，并将数据分成两个不重叠的段{0.763, 1.656, 0.424, 1.939}和{0.133, 1.881, 0.328, 1.348}。

11.14 现在对习题 11.12 的含噪数据重复习题 11.11。

11.15 现在假定习题 11.9 的原始数据被噪声高度污染，含噪数据序列变成{6.03, 6.18, 3.35, 8.42, 1.05, 7.96, 2.59, 3.75}。计算数据的能量谱，估计抽样数据的信噪比。

11.16 对习题 11.15 的数据应用巴特利法计算其改善的能量谱的估计，同样分成两个等长数据段。

11.17 对习题 11.15 的数据应用韦尔奇修正周期图法获得其高质量的能量谱估计，将数据分成三个相等长度重叠率为 50% 的数据段。

11.18 绘制一个表格，比较从习题 11.9 ~ 习题 11.17 的不同频谱估计方法和信噪比中获得的结果。讨论不同方法中信噪比的影响，选择你喜欢的方法。

11.19 从习题 11.9 的数据获取功率密度谱，即对{0, 1, 0, 1, 0, 1, 0, 1}应用布莱克曼 - 图基法，将它的结果与习题 11.9 的相比较。使用窗函数{0, 0.5, 1, 1, 1, 1, 0.5, 0}。

11.20 重复习题 11.19，使用习题 11.12 的含噪数据。

11.21 重复习题 11.19，使用习题 11.15 的含噪数据。

11.22 现在比较习题 11.12 ~ 习题 11.17 的答案，确定周期图法或布莱克曼 - 图基法是否能给出噪声条件下的最佳结果。

11.23 应用布莱克曼 - 图基法确定一个长度为 1.0 μs，幅度在 0 V 和 5 V 间变化的方波的能量谱。将你的结果与理论值进行比较。

11.24 编写计算机程序，用以产生你自己感兴趣的波形，利用本章介绍的技术设计适宜的频谱分析程序，估计能量和相位谱。

11.25 设计一些具有某些特殊性质的幅度和相位谱来考验频谱估计技术，如封闭的谱峰。将它们转换到时域，加噪得到一个较低的、一个等于 1 的和一个较高的信噪比。现在确定和评价幅度与相位谱。

参考文献

Akaike H. (1969) Fitting autoregressive models for prediction. *Ann. Institute of Statistical Mathematics*, **21**, 243–7.

Akaike H. (1973) Information theory and an extension of the maximum likelihood principle. In *2nd International Symposium on Information Theory* (Petrov B.N. and Csaki F. (eds)), pp. 267–81. Budapest: Akademiai Kiade.

Akaike H. (1974) A new look at the statistical model identification. *IEEE Trans. Automatic Control*, **19**, 716–22.

Akaike H. (1978) A Bayesian analysis of the minimum AIC procedure. *Ann. Institute of Statistical Mathematics*, **30A**, 9–14.

Akaike H. (1979) A Bayesian extension of the minimum AIC procedure of autoregressive model fitting. *Biometrika*, **66**, 237–42.

Blackman R.B. and Tukey J.W. (1958) *The Measurement of Power Spectra.* New York: Dover.

Box G.E.P. and Jenkins G.M. (1976) *Time-Series Analysis, Forecasting, and Control.* San Francisco CA: Holden-Day.

Burg J.P. (1967) Maximum entropy spectral analysis. In *Proc. 37th Meeting Society Exploration Geophysicists*, Oklahoma City, October. Reprinted in Childers D.G. (ed.) (1968) *Modern Spectrum Analysis*. New York: IEEE Press.

Burg J.P. (1968) A new analysis technique for time series data. In *NATO Advanced Study Institute on Signal Processing with Emphasis on Underwater Acoustics*, 12–23 August. Reprinted in Childers D.G. (ed.) (1968) *Modern Spectrum Analysis*. New York: IEEE Press.

Cadzow J.A. (1979) ARMA spectral estimation: an efficient closed-form procedure. In *Proc. RADC Spectrum Estimation Workshop*, Rome NY, October 1979, pp. 81–97.

Cadzow J.A. (1982) Spectral estimation: an overdetermined rational model equation approach. *Proc. IEEE*, **70**, 907–38.

Candy J.V. (1989) *Signal Processing: The Modern Approach.* New York: McGraw-Hill.

Capon J. (1969) High-resolution frequency–wavenumber spectrum analysis. *Proc. IEEE*, **57**, 1408–18.

Chatfield C. (1979) Inverse autocorrelation. *J. Royal Statistical Society A*, **142**, 363–77.

Chatfield C. (1984) *The Analysis of Time Series*, 3rd edn. London: Chapman and Hall.

Clarkson P.M. (1993) *Optimal and Adaptive Signal Processing*. Boca Raton FL: CRC Press.

Cooley J.W. and Tukey J.W. (1965) An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, **19**, 297–301.

DeFatta D.J., Lucas J.G. and Hodgkiss W.S. (1988) *Digital Signal Processing: A System Design Approach*, Section 6.6.5, p. 263. New York: Wiley.

Durbin J. (1959) Efficient estimation of parameters in moving-average models. *Biometrika*, **46**, 306–16.

Friedlander B. (1982) Lattice methods for spectral estimation. *Proc. IEEE*, **70**, 990–1017.

Gersch W. (1970) Spectral analysis of EEGs by autoregressive decomposition of time series. *Mathematical Biosciences*, **7**, 205–22.

Graupe D., Krause D.J. and Moore J.B. (1975) Identification of autoregressive-moving average parameters of time series. *IEEE Trans. Automatic Control*, **20**, 104–7.

Harris F.J. (1978) On the use of windows for harmonic analysis with the discrete Fourier transform. *Proc. IEEE*, **66**(1), 51–84.

Jenkins G.M. and Watts D.G. (1968) *Spectral Analysis and its Applications*. San Francisco CA: Holden-Day.

Jervis B.W. and Saatchi M.R. (1990) An integrated system for process control and the acquisition, storage and processing of data. In *IEE Colloq. on PC-Based Instrumentation*, 31 January, IEE Digest No. 1990/025.

Jervis B.W., Nichols M.J., Johnson T.E., Allen E.M. and Hudson N.R. (1983) A fundamental investigation of the composition of auditory evoked potentials. *IEEE Trans. Biomedical Engineering*, **30**(1), 43–50.

Jervis B.W., Coelho M. and Morgan G.W. (1989a) Spectral analysis of EEG responses. *Medical and Biological Engineering and Computing*, **27**, 230–8.

Jervis B.W., Coelho M. and Morgan G.W. (1989b) Effect on EEG responses of removing ocular artefacts by proportional EOG subtraction. *Medical and Biological Engineering and Computing*, **27**, 484–90.

Jervis B.W., Saatchi M.R., Allen E.M., Hudson N.R., Oke S. and Grimsley M. (1993) A pilot study of computerised differentiation of Huntington's disease, schizophrenia, and Parkinson's disease patients using the contingent negative variation. *Medical and Biological Engineering and Computing*, **31** (January), 31–8.

Kalouptsidis N. and Theodoridis S. (1987) Fast adaptive least-squares algorithms for power spectral estimation. *IEEE Trans. Acoustics, Speech and Signal Processing*, **35**, 661–70.

Kay S.M. (1980) A new ARMA spectral estimator. *IEEE Trans. Acoustics, Speech and Signal Processing*, **28**, 585–8.

Kay S.M. (1988) *Modern Spectral Estimation: Theory and Application*. Englewood Cliffs NJ: Prentice-Hall.

Kuo F.F. and Kaiser J.F. (1966) *System Analysis by Digital Computer*, Chapter 7, pp. 232–8. New York: Wiley.

Lacoss R.T. (1971) Data adaptive spectral analysis methods. *Geophysics*, **36**, 661–75.

Lawson C.L. and Hanson R.J. (1974) *Solving Least Squares Problems*. Englewood Cliffs NJ: Prentice-Hall.

Levinson N. (1947) The Weiner RMS error criterion in filter design and prediction. *J. Mathematical Physics*, **25**, 261–78.

Makhoul J. (1975) Linear prediction: a tutorial review. *Proc. IEEE*, **63**, 561–80.

Marple S.L. (1980) A new autoregressive spectrum analysis algorithm. *IEEE Trans. Acoustics, Speech and Signal Processing*, **28**, 441–54.

Marple S.L. (1987) *Digital Spectral Analysis*. Englewood Cliffs NJ: Prentice-Hall.

Morrison D.F. (1976) *Multivariate Statistical Methods*, 2nd edn. New York: McGraw-Hill.

Nuttall A.H. (1976) *Spectral Analysis of a Univariate Process with Bad Data Points via Maximum Entropy and Linear Predictive Techniques*. NUSC Technical Report TR-5303, New London CN.

Pardey J., Roberts S. and Tarassenko L. (1996) A review of parametric modelling techniques for EEG analysis. *Medical Engineering Physics*, **18**, 2–11.

Priestley M.B. (1981) *Spectral Analysis and Time Series*, Volume 1, *Univariate Series*, Chapters 6 and 7. New York: Academic Press.

Proakis J.G. and Manolakis D.G. (1989) *Introduction to Digital Signal Processing*, Sections 1.3.2, 11.2.4, 11.3 and 11.3.4 and Appendix 6A. Basingstoke: Macmillan.

Rissanen J. (1983) A universal prior for the integers and estimation by minimum description length. *Ann. Statistics*, **11**, 417–31.

Saatchi M.R. and Jervis B.W. (1991) PC-based integrated system developed to diagnose specific brain disorders. *Computing and Control Engineering J.*, **2**(2), 61–8.

SAS (1982) *SAS User Guide*. SAS Institute.

Shibata R. (1976) Selection of the order of an autoregressive model by Akaike's information criterion. *Biometrika*, **63**, 117–26.

Ulrych T.J. and Clayton R.W. (1976) Time series modelling and maximum entropy. *Physics Earth and Planetary Interiors*, **12**, 188–200.

Welch P.D. (1967) The use of fast Fourier transform for the estimation of power spectra. *IEEE Trans. Audio and Electroacoustics*, **15**, 70–3.

Whittle P. (1965) *Prediction and Regulation*. London: English Universities Press.

Wold H. (1954) *A Study of the Analysis of Stationary Time Series*. Stockholm: Almquist and Wichsells.

附录

11A　频谱估计和分析的 MATLAB 程序

MATLAB 信号处理工具箱包含了许多对参数法及非参数法频谱估计和分析有用的函数。我们给出下面的简单例程来说明这些函数在实际中的应用。

welchn.m　　应用韦尔奇法估计和分析频谱的说明例程；
burgm.m　　应用 Burg 法估计和分析频谱的说明例程；
yulewalkm.m　　应用 Yule-Walker 法估计和分析频谱的说明例程；

程序可以在网站上找到。应用程序的演示例子可在配套的指导手册，即 *A Practical Guide for MATLAB and C Language Implementations of DSP Algorithms*（详见前言）中找到。

第12章 通用和专用数字信号处理器

本章的主要目标是提供对通用和专用DSP处理器关键问题的一种理解，DSP算法对处理器硬件和软件体系结构的影响，以及关键DSP算法在通用数字信号处理器上的实时执行是如何实施的，或作为一个专用硬件是如何实现的。

实时通常隐含着在规定的时间限制内“尽可能快”的意思。实事处理可以分为两大类（尽管进一步的细分是可能的）：流处理（stream processing），比如数字滤波，每次处理一个数据样本；块处理（block processing），比如FFT和相关，每次处理固定数目的数据块。实时DSP算法的实现要求硬件和软件。硬件可能是处理器阵列、标准微处理器、DSP芯片或微程序控制的专用器件；软件可能是低级汇编语言代码或DSP硬件本身的微代码，也可能是高级语言，比如C或C++代码。高级语言的使用现在比较普遍，特别是对于非常复杂和先进的新型DSP处理器。

自从20世纪80年代早期引入以来，DSP处理器在复杂性和先进性方面有了显著的发展，以增强性能和应用范围。这也引起了可以使用的DSP处理器数目的显著增长。为了反映这些变化，本章包含了定点和浮点DSP处理器系列的特征和影响DSP处理器选择的因素。

12.1 引言

为了方便，DSP处理器被分成两大类：通用和专用。DSP处理器包括定点器件，比如德州仪器（Texas Instruments）的TMS320C54x和摩托罗拉（Motorola）的DSP563x处理器；以及浮点处理器，比如德州仪器的TMS320C4x和模拟器件公司（Analog Device）的ADSP21xxx SHARC处理器。

有两类专用硬件：

(1) 为特定DSP算法比如数字滤波、快速傅里叶变换的高效执行而设计的硬件。这类专用硬件有时称为特定算法（algorithm-specific）的数字信号处理器。
(2) 为特定应用比如电信、数字音频或控制应用而设计的硬件。这类硬件有时称为特定应用（application-specific）的数字信号处理器。

在大多数情况下，特定应用的数字信号处理器运行特定的算法，比如PCM编/解码，但是也要求执行其他的特定应用的操作。专用DSP处理器的例子有Cirrus的数字音频抽样率变换器（CS8420），Mitel的多通道电话声音回波对消器（MT9300），FFT处理器（PDSP16515A）和可编程FIR滤波器（VPDSP16256）。

通用和专用处理器都可以和单芯片，或者和乘法器、算术逻辑单元（ALU）、存储器等单独模块一起设计。

首先，我们将讨论已经使实时DSP在很多领域成为可能的数字信号处理器的体系结构特征。

12.2 信号处理的计算机体系结构

大多数现在可用的通用处理器都是基于冯·诺伊曼（Von Neumann）概念的，其操作是顺序执行的。图12.1显示了标准冯·诺伊曼处理器的一种简化体系结构。当指令在这样的处理器

中处理时，在每条指令阶段不参与处理的处理器中的单元（unit）处于空闲等待状态，直到将控制权传递给它们。处理器速度的增加是通过加快单个单元的操作而实现的，但是单元的操作速度是有限制的。

如果操作是实时的，DSP 处理器必须使体系结构优化以执行 DSP 函数。图 12.2 显示了一个适用于实时 DSP 的通用硬件体系结构。它的特征如下：

- 独立的数据和程序指令存储空间的多总线结构。通常数据存储器保存输入数据、中间数值和输出样本，以及用于数字滤波或 FFT 等操作的固定系数。程序指令存储在程序存储器中。
- I/O 端口提供了和外围设备（比如 ADC 和 DAC）之间传递数据，或者将数字数据传递给其他处理器的手段。如果可以使用，直接存储器访问（DMA）能够和数据RAM 直接进行数据块的快速传输，DMA 通常受外部控制。
- 用于逻辑和算术操作的算术单元，包括 ALU、硬件乘法器和移位器（或乘法累加器）。

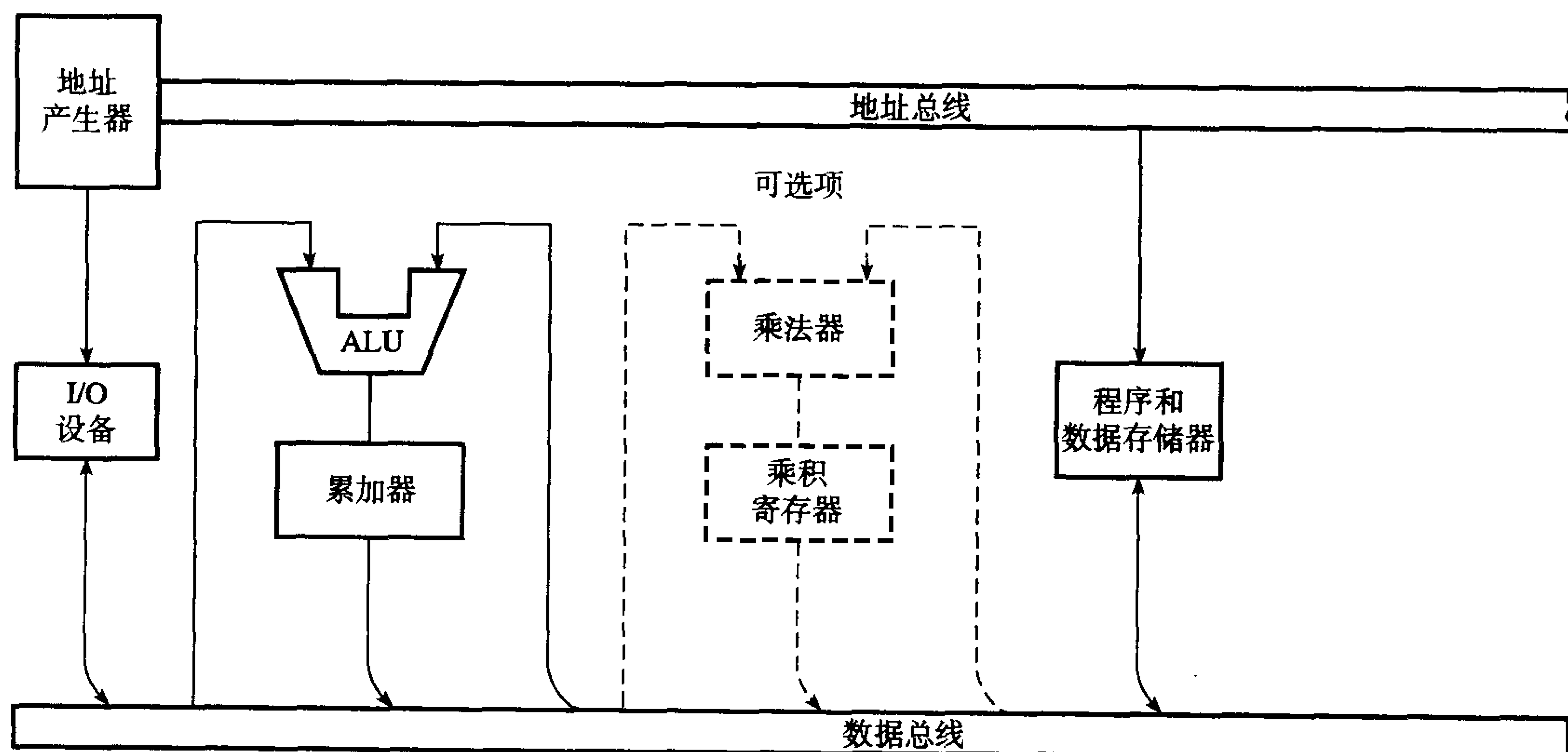


图 12.1　标准微处理器的一种简化体系结构

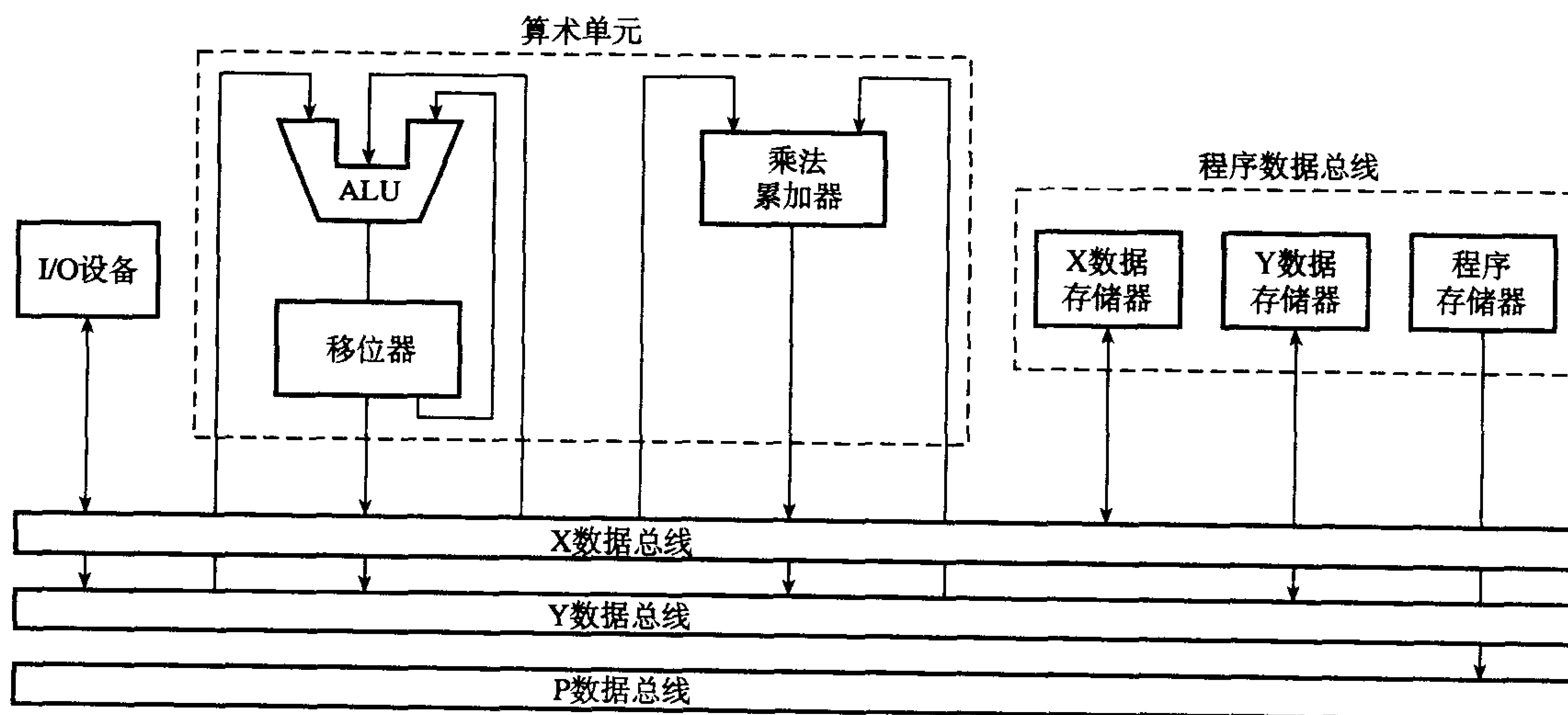


图 12.2　信号处理的基本通用硬件体系结构

为什么需要这样一种体系结构？大多数DSP算法（比如滤波、相关和快速傅里叶变换）都包含重复性的算术操作，比如乘法、加法、存储器存取以及通过CPU的大数据流。标准微处理器的体系结构不适合这类活动。DSP硬件设计的一个重要目标就是为DSP操作优化硬件体系结构和指令集。在数字信号处理器中，这是通过充分使用并行概念来实现的。特别要用到下列技术：

- 哈佛（Harvard）体系结构；
- 流水线技术（pipelining）；
- 快速的、专用的硬件乘法器/累加器；
- DSP专用指令；
- 复制技术（replication）；
- 片上存储器/高速缓存；
- 扩展的并行技术——SIMD、VLIW和静态超标量（superscalar）处理。

对于成功的DSP设计，理解这些关键的体系结构特征是非常重要的。

12.2.1 哈佛体系结构

哈佛体系结构的基本特征是程序和数据有分离的存储空间，允许指令的取指和执行完全重叠。标准微处理器（例如Intel 6502）是数据和指令单总线结构的特征，如图12.1所示。

假如在标准微处理器中，我们希望读取存储器中地址为ADR1处的数值op1，然后存储到其他两个地址ADR2和ADR3中。指令为

```
LDA   ADR1   将操作数op1从ADR1处载入累加器
STA   ADR2   将op1存入地址ADR2
STA   ADR3   将op1存入地址ADR3
```

通常每条指令包含三个清晰的步骤：

- 指令取指（fetch）
- 指令译码（decode）
- 指令执行（execute）

在我们的例子中，指令取指包含从存储器中提取下一条指令，指令执行包含读或写数据到存储器中。在标准处理器中没有哈佛体系结构，程序指令（即程序代码）和数据（操作数）保存在一个存储空间中，请参见图12.3。因此在执行当前指令时不允许提取下一条指令，因为取指和执行阶段都需要存储器存取。

在哈佛体系结构中（参见图12.4），因为程序指令和数据位于分离的存储空间中，下一条指令的提取可以和当前指令的执行重叠，参见图12.5。正常情况下，程序存储器保存程序代码，而数据存储器存储变量，例如输入数据样本。

一些数字信号处理器（例如摩托罗拉DSP56000）使用严格的哈佛体系结构，但是大部分使用改进的哈佛体系结构（例如TMS320系列处理器）。在改进的哈佛体系结构中，例如TMS320，仍旧保留了分离的程序和数据存储空间，但是允许两个存储空间之间的通信，这和严格的哈佛体系结构不一样。

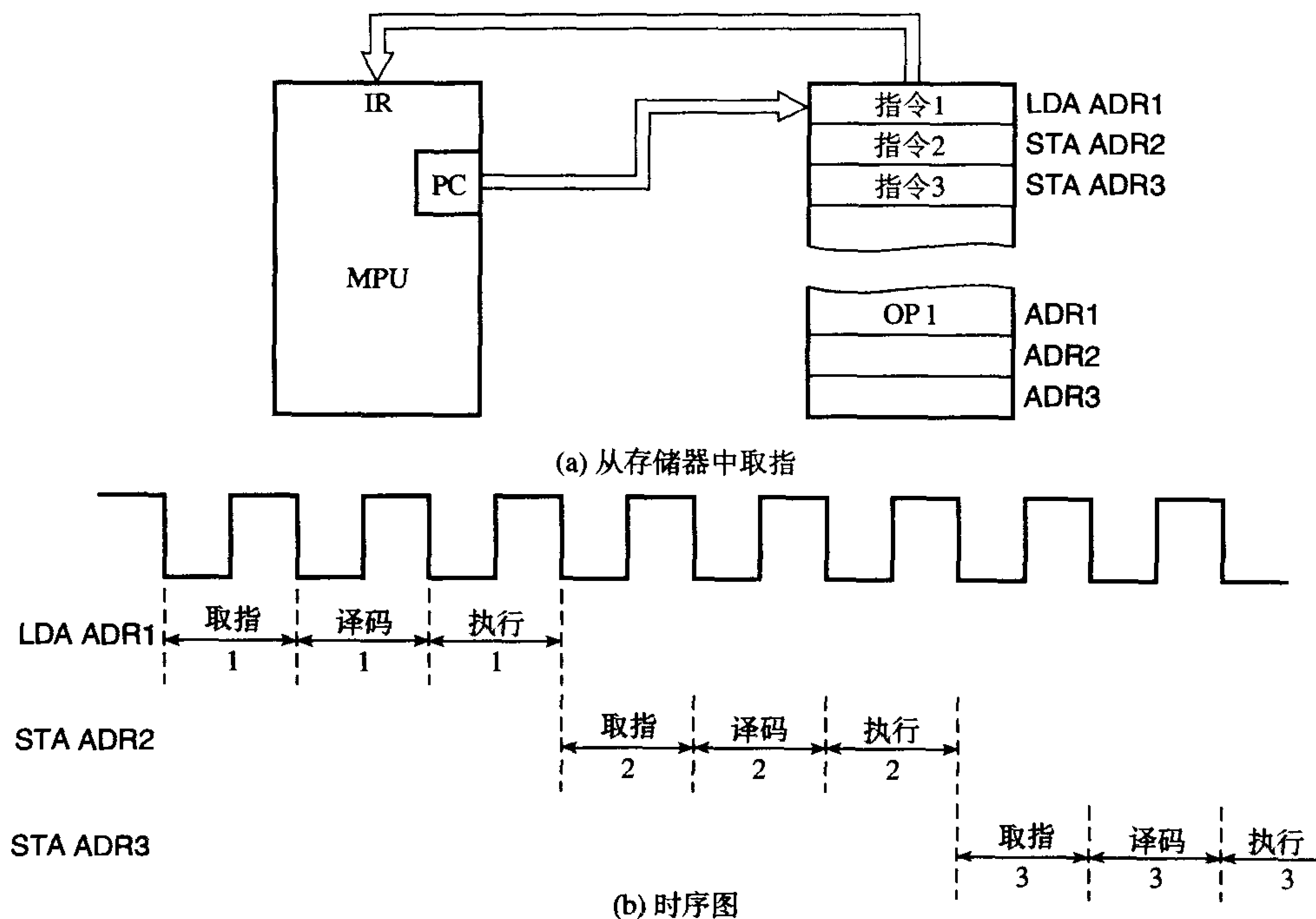


图 12.3　在单存储空间的非哈佛体系结构中指令取指、译码和执行的示例

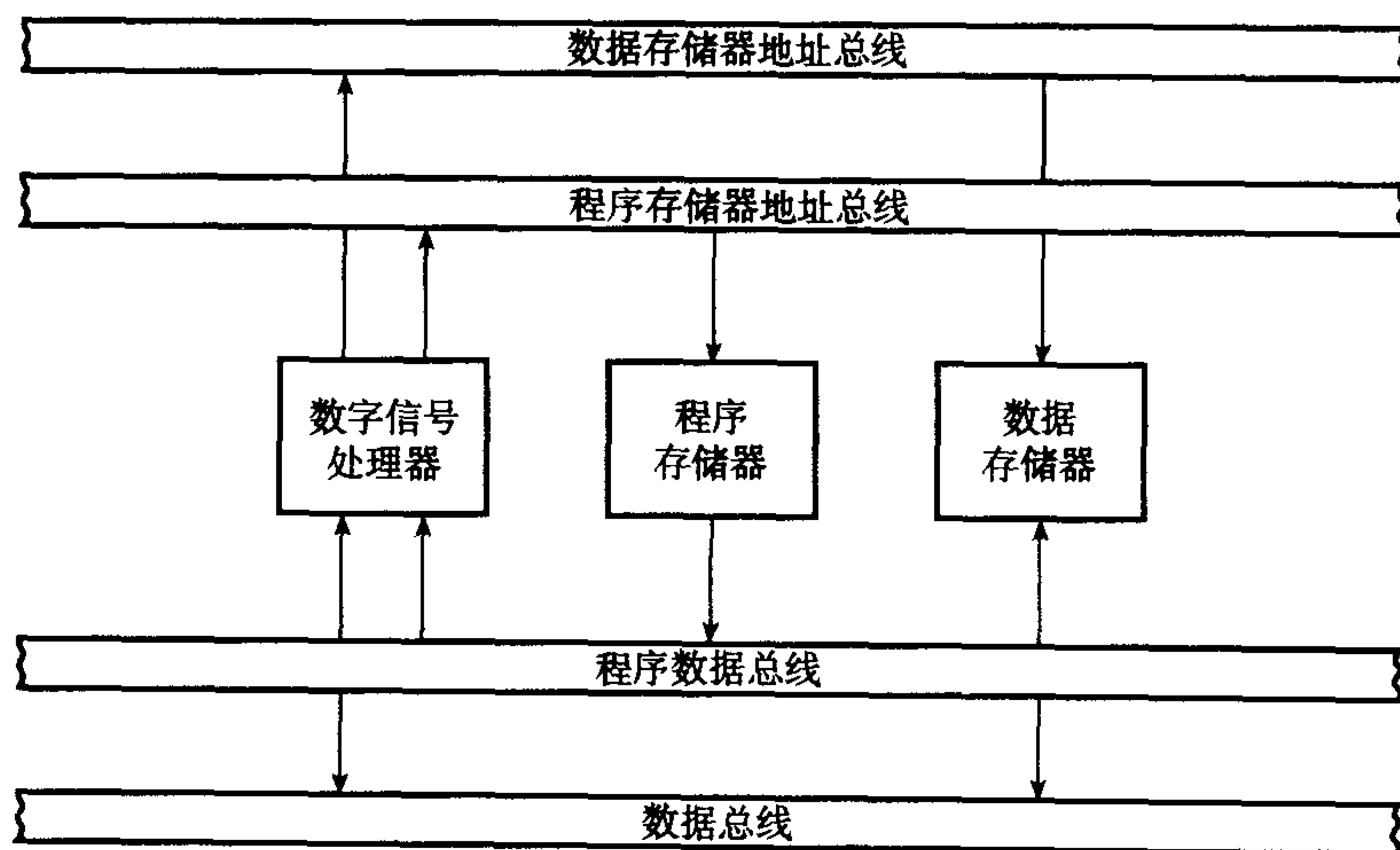


图 12.4　带有分离的数据和程序存储器空间的基本哈佛体系结构。数据和程序指令的提取可以重叠，因为使用两个独立的存储器

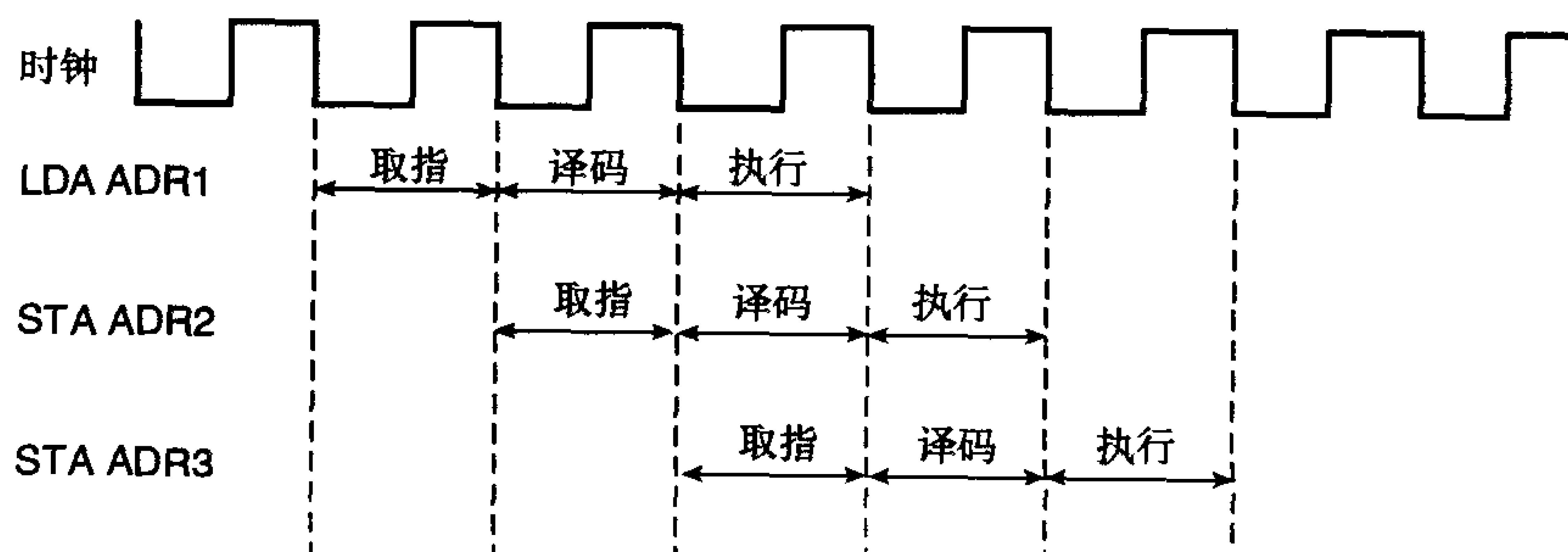


图 12.5　哈佛体系结构能够使指令重叠的示例

12.2.2 流水线技术

流水线技术允许两个或更多操作在执行时重叠。在流水线技术中，任务被分解为若干个明确的子任务，这些子任务在执行时重叠。该技术被广泛用于数字信号处理器中，以增加速度。流水线类似于工厂（比如汽车或电视机组装厂）里典型的产品线。像产品线一样，任务被分解为小的、独立的子任务，称为流水阶段（pipe stage）。这些流水阶段级联在一起形成水管（pipe），阶段是顺序执行的。

我们在上一节已经看到，一条指令可以分解为三个步骤。指令中的每个步骤可以看成流水线的一个阶段，因此可以重叠。通过重叠指令，在每个时钟周期的开始处可以开始一条新的指令（参见图 12.6(a)）。

图 12.6(b)给出了一个三级流水线的时序图，图中画出了指令的步骤。通常流水线中的每个步骤花费一个机器周期。因此在一个给定的周期中，可以同时有三条不同的指令是活动的，尽管每一条指令处于完成过程的不同阶段。指令流水线的关键是指令的三个部分（即取指、译码和执行）是独立的，这样多条指令的执行就可以重叠。在图 12.6(b)中，可以看到在第 i 个周期，处理器可以同时提取第 i 条指令，译码第$(i-1)$条指令，执行第$(i-2)$条指令。

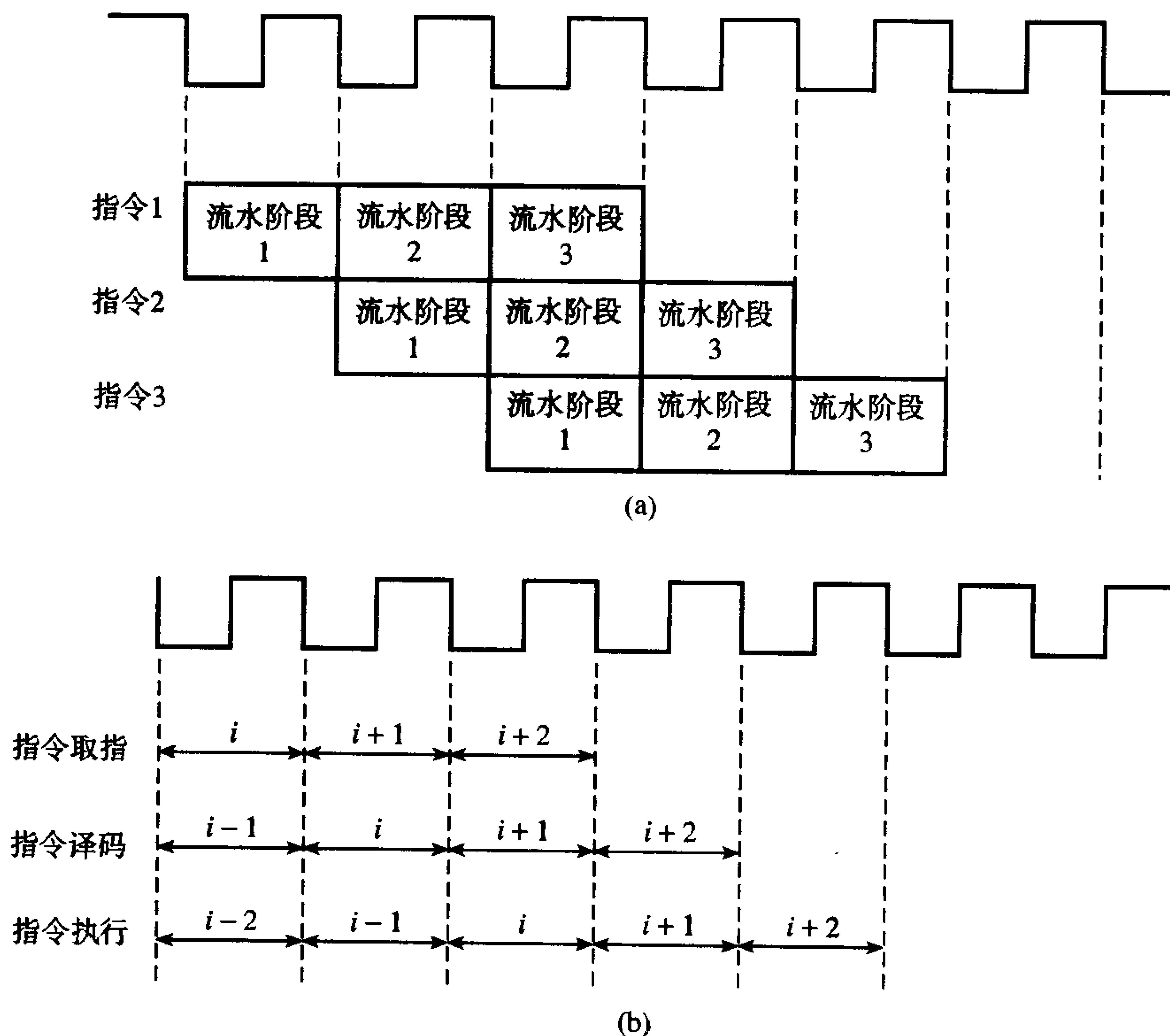


图 12.6 流水线技术概念的示例

上面讨论的三级流水线操作是基于德州仪器TMS320处理器采用的技术。和流水线技术的其他应用一样，TMS320 使用若干寄存器实现流水线：预取指计数器（prefetch counter）保存将要提取的下一条指令的地址，指令寄存器保存要执行的指令，如果当前指令仍在执行，指令队列寄存器存储要执行的指令。程序计数器包含要执行的下一条指令的地址。

通过开发指令流内在的并行性，流水线技术显著地降低了指令的平均执行时间。流水线机器的吞吐率由单位时间内通过水管的指令数决定。和产品线一样，流水线的所有阶段必须是同步的。将

指令在水管中从一个步骤移到另一个步骤的时间（参见图12.6(a)）是一个周期、并且依赖于流水线中最慢的阶段。在完美的流水线中，指令的平均时间由下式给出（Hennessy and Patterson, 1990）

$$\frac{\text{每条指令的时间（非流水线）}}{\text{流水阶段的数目}} \tag{12.1}$$

在理想情况下，增加的速度等于流水阶段的数目。实际上，由于设置流水线的开销（overhead）、流水线寄存器的延迟等影响，增加的速度会小于流水阶段的数目。

例 12.1　在非流水线机器中，指令的取指、译码和执行分别花费 35 ns、25 ns 和 40 ns。如果指令步骤流水线化，确定吞吐率的增加量。假定每个阶段流水线的开销是 5 ns，并且忽略其他延迟。

解：

在非流水线机器中，平均指令时间是所有步骤的执行时间的简单求和：35 + 25 + 40 ns = 100 ns。然而，如果我们假定处理器有固定的机器周期，并且指令步骤同步于系统时钟，那么每条指令将花费 3 个机器周期来完成：40 ns × 3 = 120 ns。这对应的吞吐率是每秒 8.3×10^6 条指令。

在流水线机器中，时钟速度由最慢阶段的速度加上流水线开销决定。在我们的例子中，机器周期是 40 + 5 = 45 ns。这限制了平均指令执行时间。吞吐率是每秒 22.2×10^6 条指令（当流水线满时）。因此

$$\begin{aligned}\text{加速} &= \frac{\text{平均指令时间（非流水线）}}{\text{平均指令时间（流水线）}}\\ &= 120/45 \\ &= 2.67\text{倍（假定非流水线执行3个周期）}\end{aligned} \tag{12.2}$$

在流水线机器中，每条指令仍花费3个时钟周期，但是在每个周期处理器执行3条不同的指令。流水线技术增加了系统吞吐率，但并不减少每条指令本身的执行时间。通常，由于流水线的开销，每条指令的执行时间会有少量增加。

流水线技术对系统存储器有显著影响。在流水线机器中，存储器存取次数的增加主要由阶段的数目决定。在 DSP 中，使用哈佛体系结构，其中数据和指令位于分离的存储空间，提升了流水线技术的性能。

当低速单元（比如数据存储器）和算术元件级联时，算术单元经常为了等待数据而空闲很长的时间。在这种情况下，使用流水线技术能够更好地利用算术单元。下一个例子解释了这个概念。

例 12.2　大多数 DSP 算法由下面公式代表的乘法－累加操作表征：

$$a_0x(n) + a_1x(n-1) + a_2x(n-2) + \ldots + a_{N-1}x(n-(N-1))$$

图 12.7 显示了一个执行上面公式的算术元件的非流水线配置。假定存储器、乘法器和累加器的传输延迟分别是 200 ns、100 ns 和 100 ns。

(1) 系统吞吐率是多少？

(2) 用流水线技术重新配置系统，使速度增加为 2∶1。

用时序图解释新配置的操作过程。

解：

(1) 系数 a_k 和数据数组存储在存储器中，如图 12.7 所示。在非流水线模式中，系数和数据依次访问，送给乘法器。乘积在累加器中求和。连续的乘法－累加操作（MAC）每 400 ns（200+100+100）执行一次，给出每秒 2.5×10^6 次操作的吞吐率。

(2) 算术操作可以分解为三个明确的步骤：读存储器、乘法和累加。为了提高速度，这些步骤可以重叠。2 : 1 的速度改善可以通过在存储器和乘法器之间、乘法器和累加器之间插入流水线寄存器而实现，如图 12.8 所示。图 12.9 是流水线配置的时序图。在时序图中可以明显看出，MAC 每 200 ns 执行一次。限制因素是通过最慢元件（在本例中是存储器）的基本传输延迟。流水线开销已经被忽略。

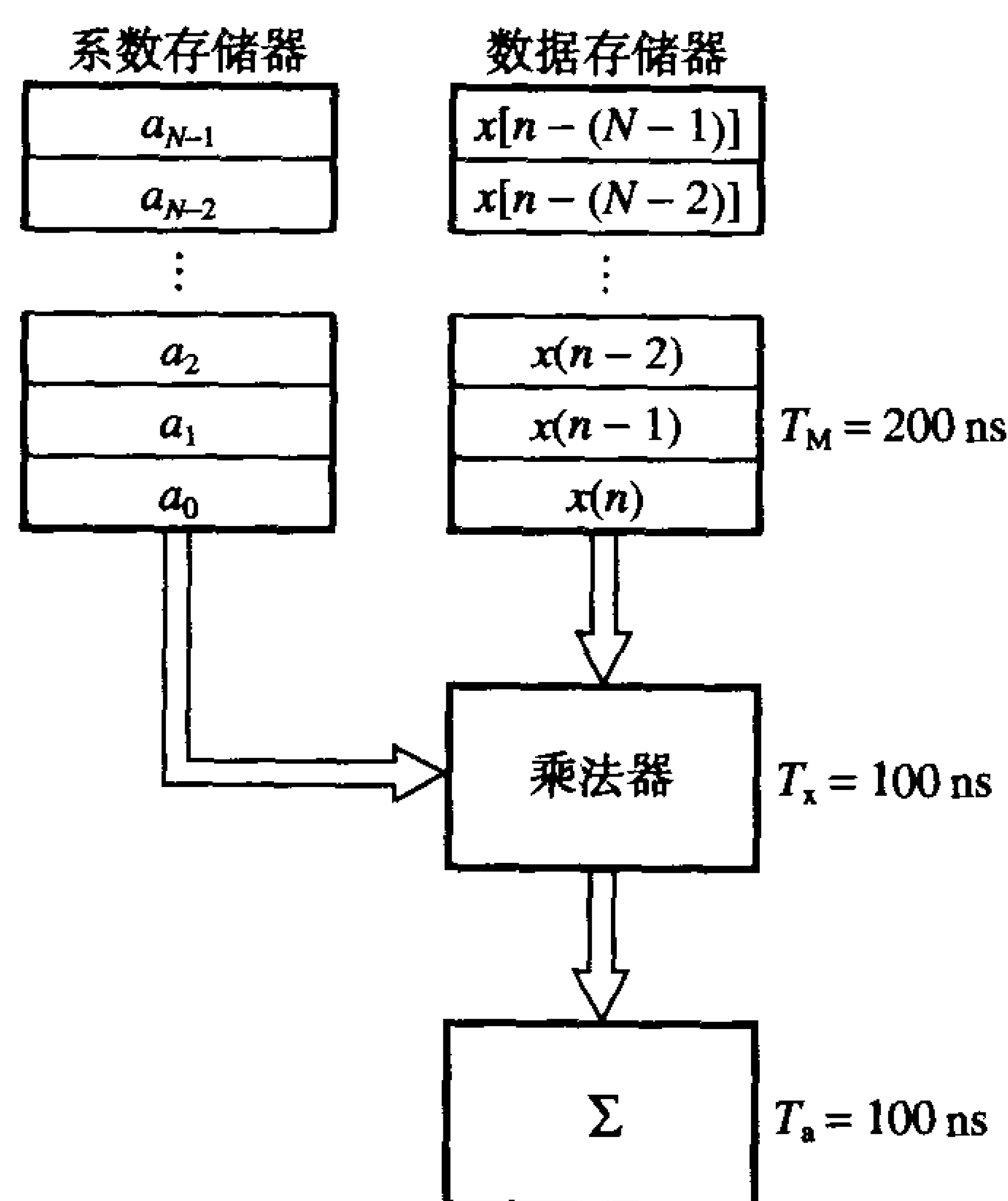


图 12.7　非流水线 MAC 配置。乘积以 400 ns 的时钟进入累加器

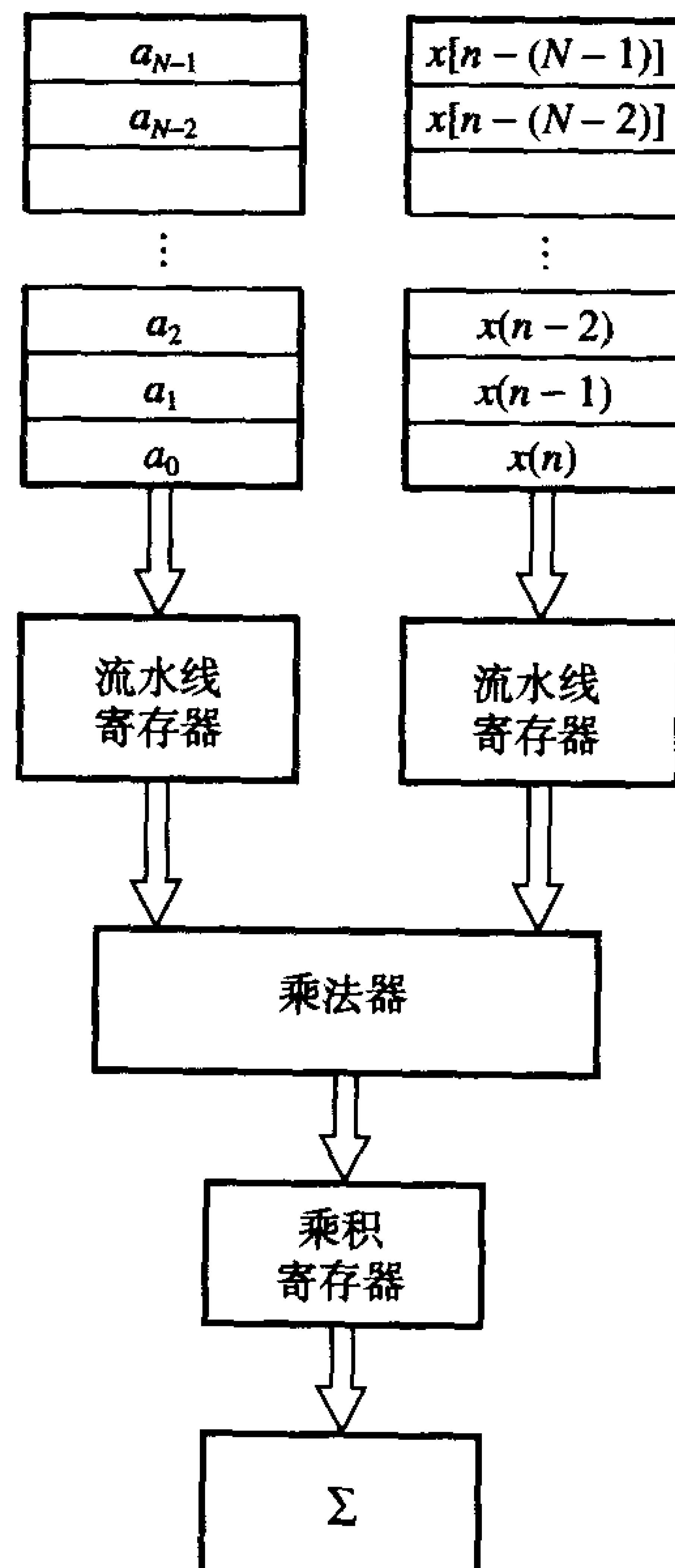


图 12.8　流水线MAC配置。流水线寄存器作为系数和数据样本的临时存储器。乘积寄存器也作为乘积的临时存储器

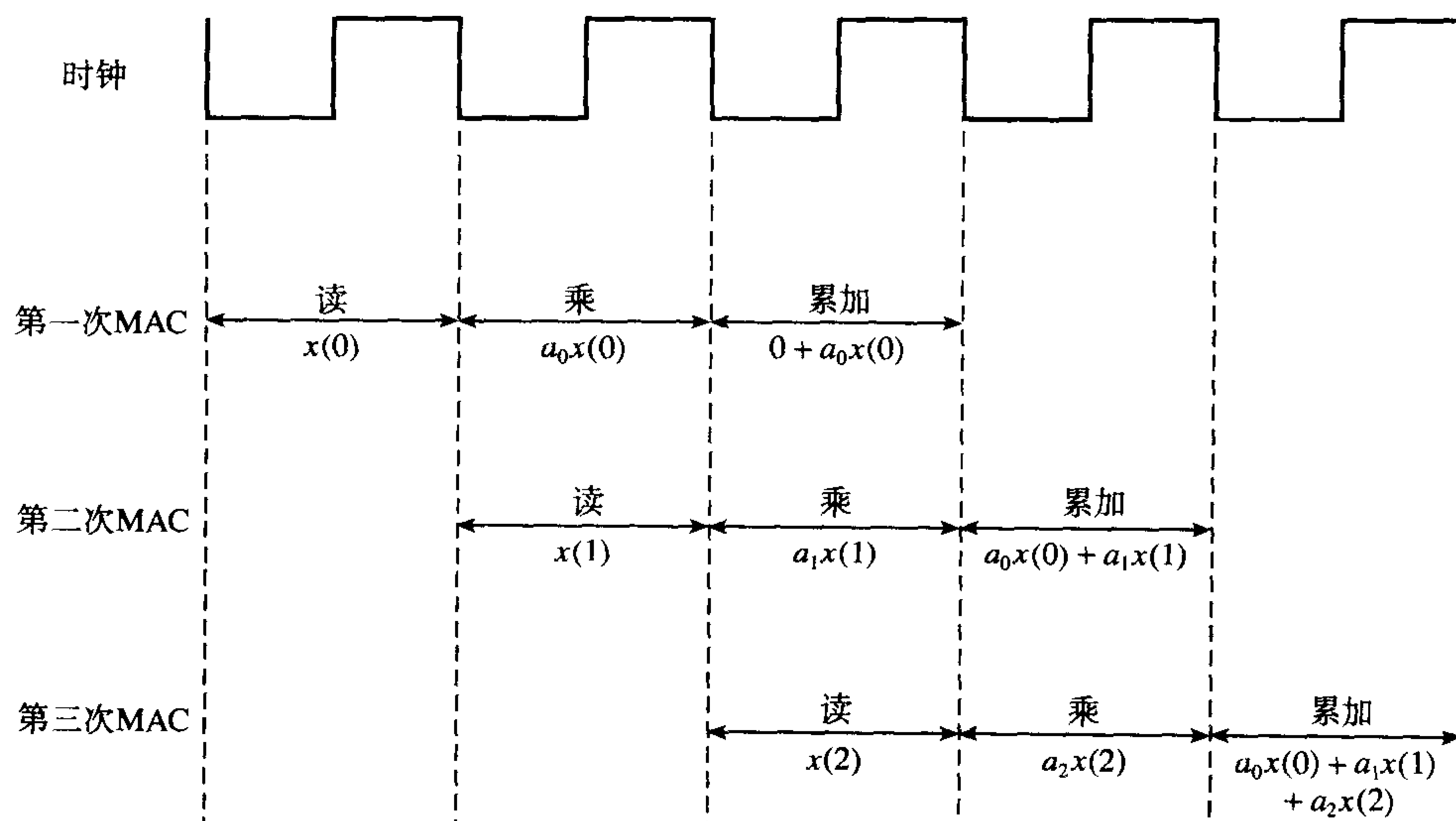


图 12.9　流水线 MAC 单元的时序图。当流水线满时，MAC 操作每个时钟周期（200 ns）执行一次

DSP算法经常是重复性但高度结构化的，非常适合多级流水线操作。例如，FFT要求连续的蝶形（butterfly）计算。尽管每个蝶形计算要求不同的数据和系数，基本的蝶形算术操作是相同的。因此算术单元例如 FFT 处理器可以改进以利用这个优点。流水线技术保证了 CPU 稳定的指令流，一般能带来系统吞吐率的显著增加。然而，有时流水线技术也可能会产生一些问题。例如，在某些数字信号处理器中，流水线技术可能会导致不需要的指令被执行，特别在分支指令（branch instruction）附近，设计人员应该意识到这种可能性。

12.2.3　硬件乘法 - 累加器

DSP中的基本数值操作是乘法和加法。众所周知，乘法在软件上是非常耗时的。如果使用浮点算术，加法甚至更耗时。为了使实时DSP成为可能，使用定点或浮点算术的快速、专用硬件乘法 - 累加器（MAC）是非常必要的。定点或浮点硬件 MAC 现在是所有数字信号处理器的标准配置。在定点处理器中，硬件乘法器通常接收两个 16 位的、小数部分用 2 的补码表示的数值，在单个周期（通常是 25 ns）内计算出一个 32 位的乘积。通过使用特殊的重复指令，平均 MAC 指令时间能够显著减少。

图 12.10 描述了一个典型的 DSP 硬件的 MAC 配置。在这个配置中，乘法器有一对输入寄存器保存乘法器的输入，一个 32 位的乘积寄存器保存乘法的结果。P（乘积）寄存器的输出连接到一个双精度累加器，在那里乘积进行累加。

硬件浮点乘法 - 累加器的原理是非常相似的，除了输入和输出规范化为浮点数值。浮点 MAC 能够以最小的误差快速计算 DSP 的结果。在第 7 章和第 8 章中讨论过，DSP 算法（比如 FIR 和 IIR 滤波）受有限字长效应的影响（系数量化和算术误差）。浮点表示提供了较宽的动态范围，减少了算术误差，尽管对很多应用来说定点表示提供的动态范围已经足够。

12.2.4　特殊指令

数字信号处理器提供了针对 DSP 优化的特殊指令。这些特殊指令的好处体现在两个方面：使代码更加紧凑从而占据更少的存储空间（几乎和使用高级语言比如 C 编写的代码一样紧凑）；使

DSP算法的执行速度增加。DSP芯片提供的特殊指令包括: (i)支持基本DSP操作的指令, (ii)减小指令循环开销的指令, (iii)面向应用的指令。

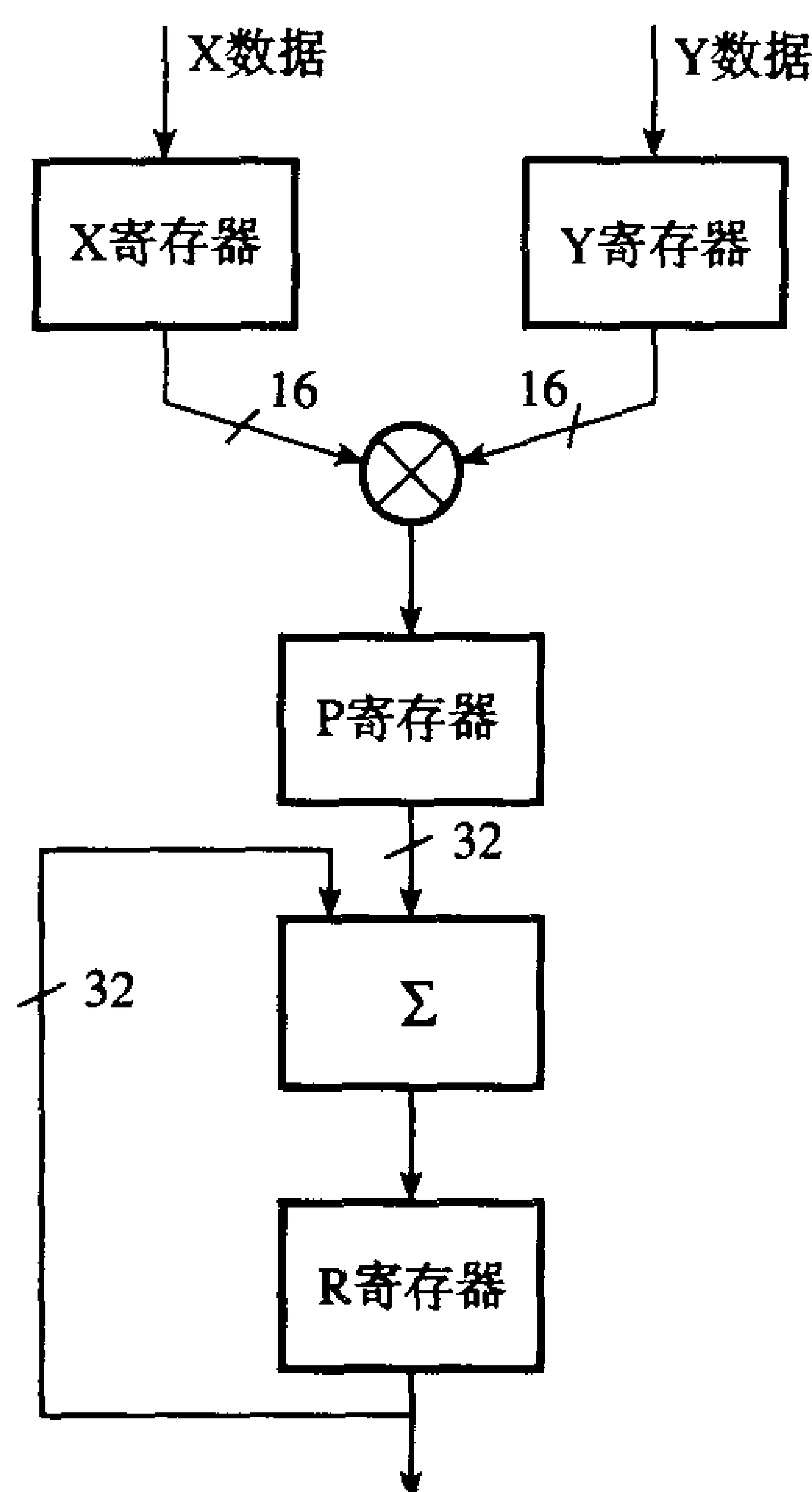


图 12.10 DSP 中典型的 MAC 配置

DSP的很多关键算法，比如数字滤波和相关，要求数据移位或延迟以便为新的数据样本腾出空间。数字信号处理器提供的特殊指令，允许当数据样本正在从存储器中提取或操作时，将其复制到下一个更高的存储器地址，所有这些操作都在一个周期内完成。例如，在第二代TMS320系列DSP处理器中，指令对LTD和MPY，允许同时进行数据装载（将数据载入乘法器的临时寄存器）、数据移位（实现符号z^{-1}表示的单位延迟）和乘积累加。支持带数据移位的乘法和累加（MAC）操作的特殊指令现在是现代DSP处理器的标准配置。

还有一些特殊指令用于加快经常重复的DSP操作的速度。例如，在第二代TMS320系列DSP处理器中，重复指令能够使下一条指令重复规定的次数。因为重复指令只要求单次取指，正常情况下要求多周期循环的代码，就可以有效地简化为单周期指令。重复指令对于DSP例如FIR和自适应滤波中的内循环计算，以及数据移动例如FFT中的倒位（bit reversal）是特别有用的。

回想FIR滤波器是由以下方程表征的：

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k)$$

其中N是滤波器的长度。

例如，在TMS320C50中，FIR方程可以用下列指令高效地实现：

```
RPT     NM1
MACD    HNM1, XNM1
```

第一条指令，RPT NM1，将减1后的滤波器长度（N−1）载入重复指令计数器，使它后面的带数据移动的乘法－累加（MACD）指令重复N次。MACD指令在一个周期内执行若干操作：

(1) 将数据存储器中的数据样本$x(n-k)$和程序存储器中的系数$h(k)$相乘；

(2) 将前一个乘积加到累加器。

(3) 实现符号 z^{-1} 表示的单位延迟，通过移动数据样本 $x(n-k)$直到更新抽头延迟线。

在摩托罗拉DSP56000 DSP处理器系列和TMS320系列中，MAC指令和重复指令（REP）一起，可以用来高效地实现FIR滤波器：

```
REP     #N–1
MAC     X0, Y0, A     X: (R0)+, X0     Y: (R4)+, Y0
```

这里，重复指令和MAC指令一起用来执行相同的乘法和乘积求和操作。再次注意到一条指令执行多次操作的能力，通过多条数据通道就可以实现。寄存器X0和Y0的内容一起相乘，乘积加到累加器。同时，从存储器X和Y中为乘法运算提取下一个数据样本和对应的系数。

在大多数现代DSP处理器中，能够使代码块而不只是单条指令重复规定次数的特殊指令进一步采用了指令重复的概念。在TMS320系列中（例如TMS320C50、TMS320C54和TMS320C30），零循环开销的指令块重复执行的格式是

```
RPTB loop
      :
      :
loop  （最后一条指令）
```

某些DSP处理器提供的重复指令具有高级语言的特征。在摩托罗拉DSP56000和DSP56300系列中，提供了可以嵌套的零开销的DO循环。下面的例子是一个嵌套的DO循环，外循环执行N次，内循环执行NM次。

```
      DO #N, LOOP1
      :
      DO #M, LOOP2
      :
      :
LOOP2（最后一条指令放在这里）
      :
LOOP1（外循环的最后一条指令放在这里）
```

嵌套循环对高效地实现DSP函数（比如FFT算法）和二维信号处理是非常有用的。

模拟器件公司的DSP处理器（例如ADSP-2115和SHARC处理器）也有嵌套循环能力。ADSP-2115支持多达四级嵌套循环。循环格式为

```
      CNTR = N
      DO LOOP UNTIL CE
      :
      :
LOOP:（循环的最后一条指令）
```

循环一直重复直到计数器到达设定的次数。循环能够包含大块的指令，而不只是单个指令。嵌套循环的格式基本上同DSP56000系列相同。

自适应滤波是现代信号处理的另一个关键的基本函数，有特殊指令来支持它。在自适应滤波中，系数更新是一个重要的步骤（参见第10章），包含根据前面的系数值计算一组新值。例如，在基于LMS算法的自适应滤波器中，系数更新任务由下列公式表征：

$$h_{k+1}(i) = h_k(i) + 2\mu e_k x(k-i) \tag{12.3a}$$

其中

$$e_k = y_k - \hat{y}_k \tag{12.3b}$$

$\hat{y}_k$ 是自适应滤波器的输出，

$$\hat{y}_k = \sum_{i=0}^{N-1} h_k(i)x(k-i) \tag{12.3c}$$

通常，系数更新任务的FIR滤波部分（12.3c式）使用前面描述的特殊MAC指令实现。系数更新任务（12.3a式）可以使用零循环开销的块重复指令执行。在TMS320C54中，基于LMS的自适应滤波器指令LMS、乘法指令ST//MPY以及指令块重复指令RPTBD，可以一起用来计算自适应滤波器的输出和更新滤波器系数。这可以减少自适应滤波器的执行时间和代码大小。

在另一个关键的DSP函数——快速傅里叶变换中，总是需要在FFT之前搅乱（scrambling）输入数据的顺序，在FFT之后再归整（unscrambling），以保证数据以正确的顺序显示。所有高性能的通用DSP处理器都提供了用于倒位寻址（bit-reversed addressing）的特殊指令，在数据样本移动或提取的同时，执行要求的搅乱/归整。

例如，TMS320的倒位寻址能够用于在存储N点复输入数据序列的同时，执行倒位：

```
RPT     N2
BLDD    #XN, *BR0+
```

在此例中，输入数据是复数，因此每个数据样本由一个实数和一个虚数数值组成。因此，每个数据样本存储在两个数据位置。

现代DSP处理器的另一个特征是具有面向应用的指令，用于语音编码(例如编码簿(codebook)搜索)、数字音频（例如环绕声）以及电信（例如维特比（Viterbi）解码）等应用。

12.2.5 复制

在DSP中，复制包含使用两个或更多的基本单元，例如使用一个以上的ALU、乘法器或存储器单元。通常安排这些单元同时工作。在DSP中，标准是一个CPU，带一个或更多复制的算术元件。

然而，成熟的并行处理概念现在正扩展到DSP，例如很多独立的处理器执行一个给定任务，或几个处理器在一个控制单元的控制下同时解决单个问题。许多并行DSP处理器，例如TMS320C40和ADSP-21060 SHARC，现在已经可以买到。

12.2.6 片上存储器/高速缓存

在大多数情况下，DSP芯片的操作很快以至于慢速的廉价存储器跟不上。通用的实际做法是加入等待状态使处理器速度降低。在一些处理器中，等待状态是软件可编程的；但在另外一些处理器中，需要一片外部硬件来降低处理器速度。等待状态当然意味着处理器不能全速工作。

为了减轻这个问题的影响，很多DSP芯片含有快速的片上数据RAM和/或ROM。在这些处理器中，低速外部存储器可以用来保存程序代码。在初始化时，可以将代码传输到快速的内部存储器以全速执行。快速的片上EPROM对实时开发和最终原型是很有用的。一些芯片提供了一块片上程序高速缓存，可以用来保存经常重复的程序段。在高速缓存中执行代码避免了进一步的存储器提取，加速了程序的执行。

提供片上存储器现在是一个标准要求。

12.2.7 扩展的并行技术——SIMD、VLIW和静态超标量处理

DSP处理器体系结构设计的趋势是增加每个周期执行的指令数和每条指令执行的操作数，以增强性能（Hacker, 1999; Texas Instruments, 1999; Berkeley Design Technology, 1999; Levy, 1999; Blalock, 1997）。

在新型DSP处理器体系结构中，广泛使用并行处理技术以实现计算性能的增强。用到的三种技术通常组合使用，即单指令多数据（single instruction, multiple data, SIMD）、超大指令字（very-large-instruction-word, VLIW）和超标量处理（superscalar processing）（Hacker, 1999; Texas Instruments, 1999; Hayes, 1998）。

SIMD处理用来增加每条指令执行的操作数。通常，在SIMD体系结构的DSP处理器中，处理器有多条数据通道和多个运行单元。因此，单个指令可以发布给多个执行单元，同时处理几块数据，使用这种方法就增加了一个周期内执行的操作数（例如图12.11）。在图12.11的例子中，DSP处理器有两个执行单元，每个都有自己的ALU、MAC和移位器。使用单条指令，处理器能够同时执行两个独立的算术操作（例如加法和MAC）。具有SIMD体系结构和双执行单元的处理器包括朗讯科技（Lucent Technologies）的DSP16000、德州仪器的TMS320C62x、模拟器件公司的TigerSHARC和ADSP-TS001。

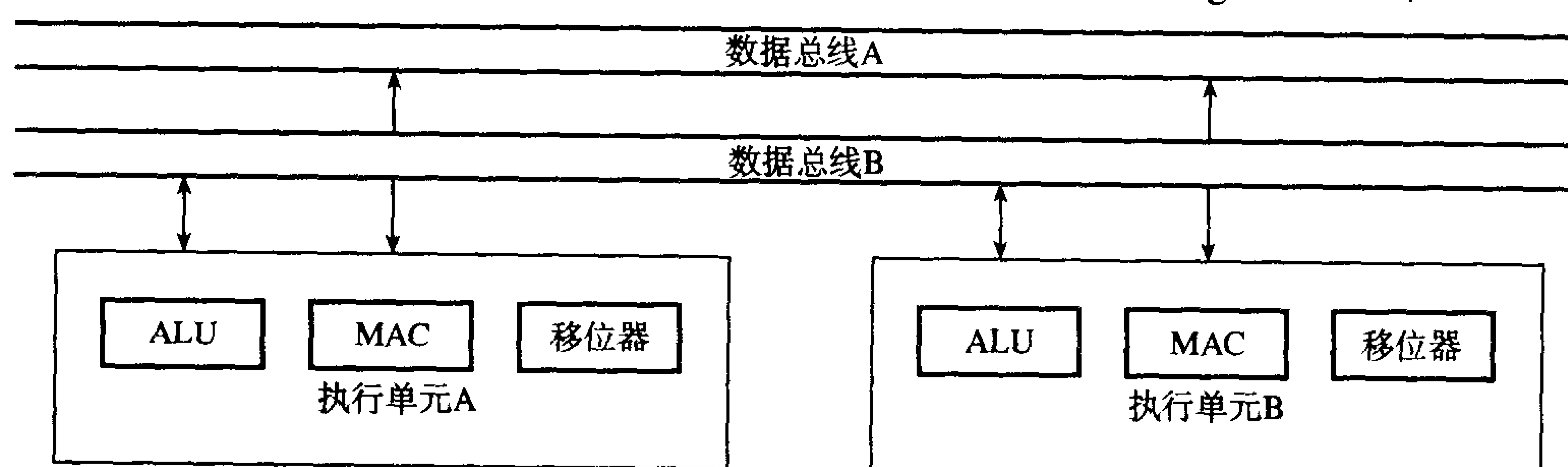


图 12.11　用于SIMD处理的带双数据通道的双算术单元

SIMD体系结构，特别是那些支持多种数据大小的SIMD体系结构的一个诱人特征是能够有效地增加可用的执行单元的数量，然后通过将这些执行单元分区，能够增加每个周期执行的操作数。例如，在TigerSHARC处理器中，两个乘法器的每一个都可以被分开，以同时执行四个16位 × 16位的乘法 - 累加操作，代替一个32位 × 32位的乘法 - 累加操作（参见图12.12）。

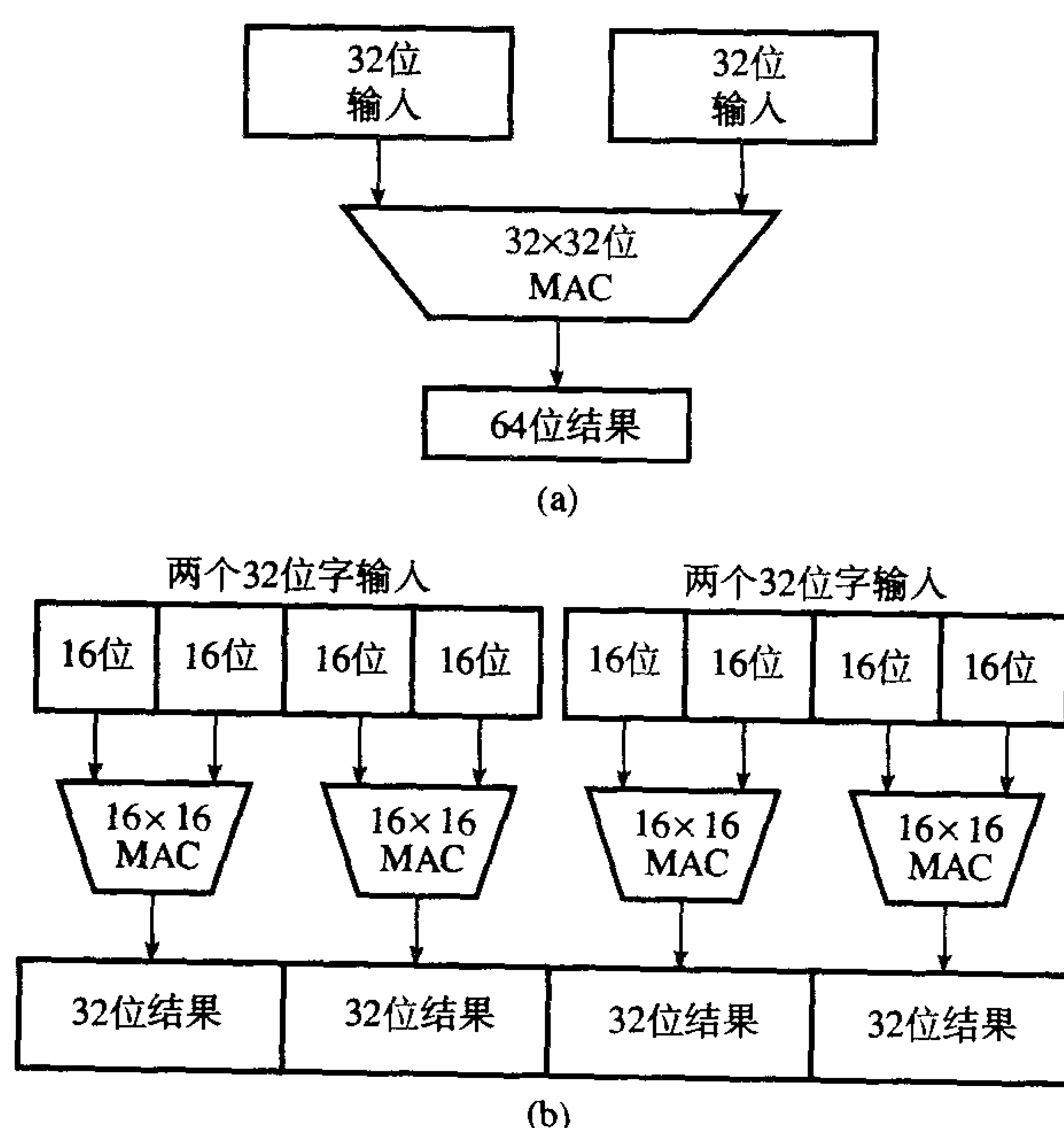


图 12.12　在TigerSHARC处理元件中，使用SIMD处理和多种数据大小的功能将乘法/累加器（MAC）数量从一个扩展到四个的例子

显然，在数据并行处理的应用中，SIMD处理能够显著地增强处理器性能。然而，在数据顺序处理的应用中，SIMD处理增加的计算性能的范围比较小。正是因为这个原因，瞄准多通道应用——比如第三代移动通信系统的下一代DSP处理器的趋势是具有SIMD能力（Hacker, 1999; Texas Instruments, 1999; Levy, 1999）。

超长指令字（VLIW）处理是一种能够显著增加每个周期处理的指令数的重要方法（Texas Instruments, 1999）。一个超长指令字在本质上是几个短指令的串连，要求多个执行单元，并行运行，从而在单个周期内执行这些指令。

图12.13是高级定点DSP处理器TMS320C62x系列的VLIW体系结构和数据流的基本原理。CPU包含两条数据通道和八个独立的执行单元，组成为两组——(L1，S1，M1，D1)和(L2，S2，M2，D2)。在本例中，每个短指令是32位宽的，八个这样的短指令链接在一起形成一个可以并行执行的超长指令字包。

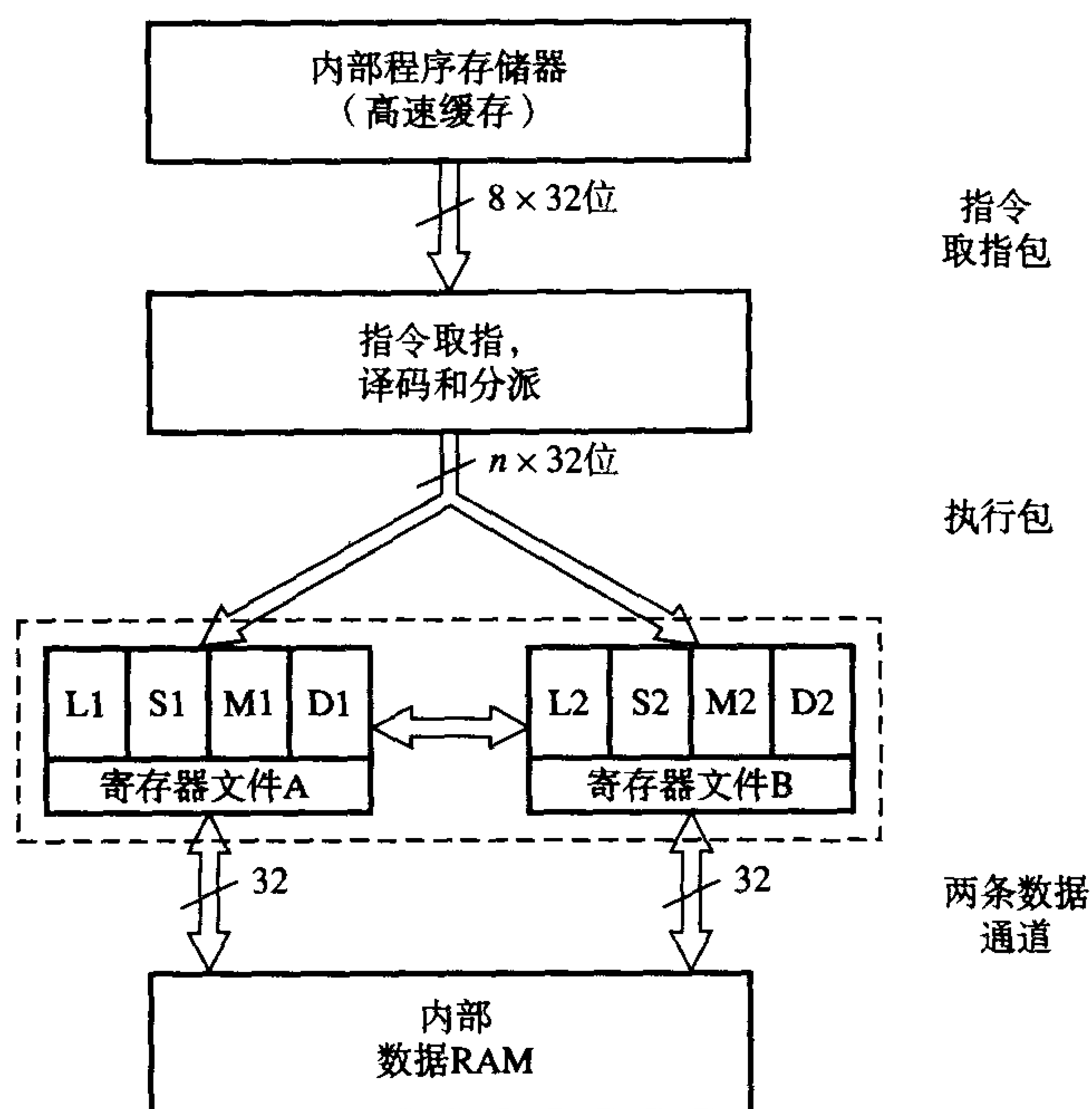


图12.13 TMS320C62x中的超长指令字（VLIW）体系结构和数据流的基本原理。注意：L1、L2是逻辑单元；S1、S2是移位器/逻辑单元；M1、M2是乘法器；D1、D2是数据寻址单元

VLIW处理当CPU从片上程序存储器提取一个指令包（八个32位的指令）的时候开始。取指包中的八条指令形成一个执行包（如果它们可以并行执行），然后适当地分派给八个执行单元。当执行包在译码和执行时，从程序存储器中提取下一个256位的指令包。如果取指包中的八条指令不能并行执行（例如，如果八条指令都是乘法-累加指令，那么只有两条指令能够在一个周期内执行，因为只有两个乘法器可用），那么就会形成几个执行包分派给执行单元，每次一个。取指包总是256位宽（八条指令），但是执行包宽度可以在一条和八条指令之间变化。

显然，VLIW体系结构是设计用来支持指令级的并行性的。这种体系结构和快速时钟速度（通常是200 MHz）一起，引导出超高性能的DSP处理器。在TMS320C62x中，指令并行性在编译阶段规划。然而，如果指令不能并行执行，这些处理器的计算效率将会下降。

超标量处理是另一种通过开发指令级的并行性以增加DSP处理器指令速率（一个周期内处理的指令数）的技术。传统上，术语"超标量"指能够在一个周期内执行多条指令的计算机体系结构

（Hayes, 1998）。这种体系结构广泛用于通用处理器，比如 PowerPC 和奔腾处理器。超标量 DSP 处理器提供了多个执行单元，几个指令可以发布给这些单元并发执行。也可以广泛使用流水线技术以进一步增加性能。

最著名的超标量 DSP 处理器是模拟器件公司的 TigerSHARC（Hacker, 1999），请参见图 12.14。TigerSHARC 被描述为一个静态超标量 DSP 处理器，因为指令中的并行性是在运行前确定的。事实上，TigerSHARC 处理器组合了 SIMD、VLIW 和超标量的概念。这个高级的 DSP 处理器具有多条数据通道和两组独立的执行单元，每个单元都带有一个乘法器、ALU、一个 64 位移位器和一个寄存器文件（参见图 12.14）。TigerSHARC 是一个浮点处理器，但是它也支持多种数据类型（8 位、16 位和 32 位数值）的定点算术。

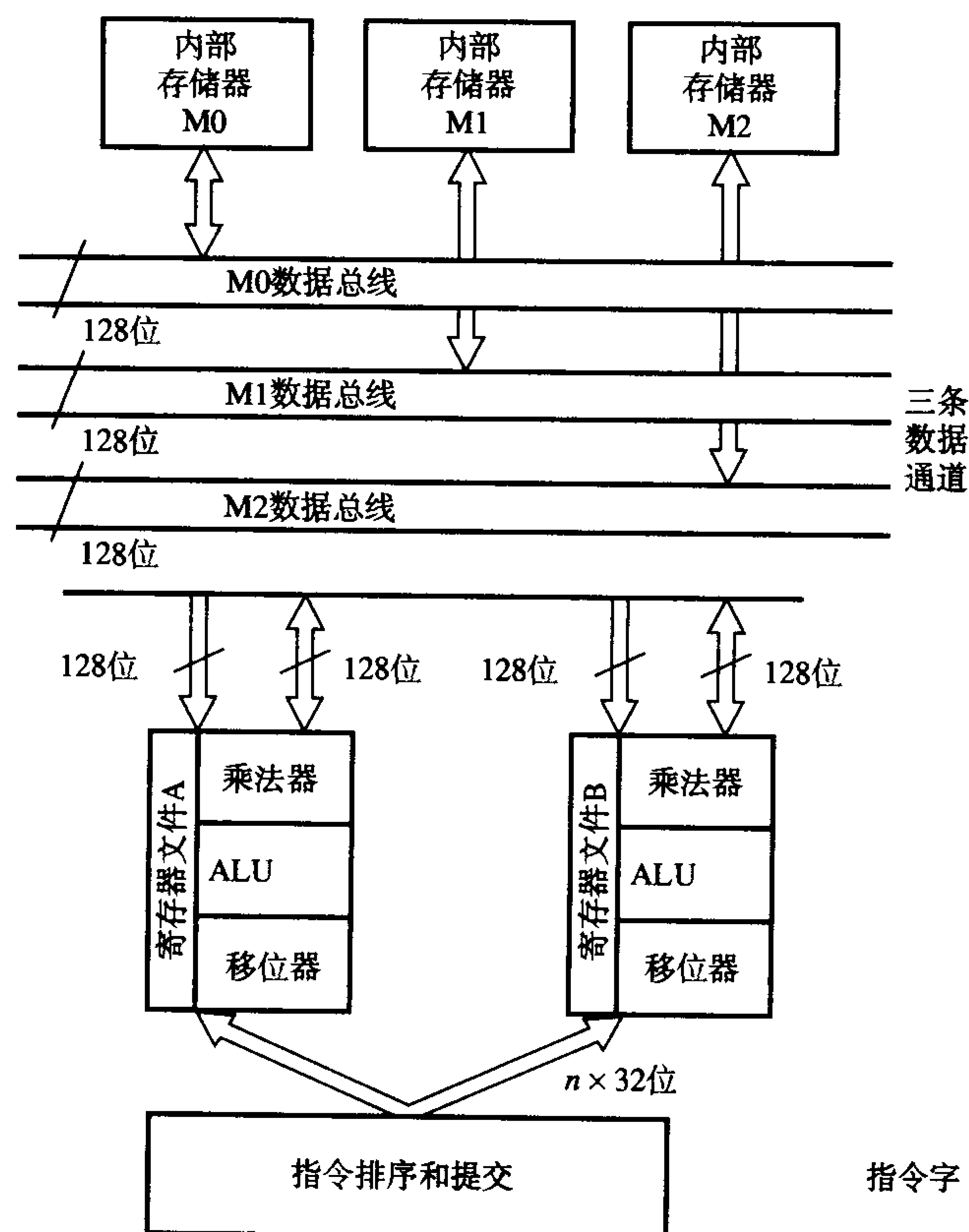


图 12.14　TigerSHARC DSP 处理器中的超标量体系结构和数据流的基本原理

指令宽度在 TigerSHARC 中不是固定的。在每个周期，可以从内部程序存储器提取多达四个 32 位的指令，提交给两组并行的执行单元。指令可以并行地提交给两个单元（单指令，多数据；即 SIMD 指令），或者独立地提交给每个单元。每个执行单元（ALU、乘法器或移位器）从寄存器文件中获取输入，并且将结果返回到寄存器文件中。寄存器文件连接了三条数据通道，所以能够在一个周期内同时从存储器读取两个输入和写入一个输出。这种装载/存储体系结构非常适合于通常获取两个输入计算一个输出的基本 DSP 操作。

正如早先讨论的，因为处理器能够工作于几种数据大小（8 位、16 位、32 位和 64 位），所以执行单元允许进一步的并行计算。所以，在每个周期中 TigerSHARC 能够执行多达八个加法/减法操作和八个 16 位输入的乘法－累加操作，以代替两个 32 位输入的乘法－累加操作。处理混合数据字节和为执行单元将大指令字分解成独立指令的能力，使得处理器能够充分开发指令级的并行性。

值得指出的是，在使用高级体系结构（比如VLIW）和超标量的DSP处理器中，在运行前对指令进行某种形式的静态规划，对于有效地使用并行处理单元将是非常必要的。避免与数据依赖性（例如在准备好之前就需要的结果）和控制依赖性（例如分支指令）相关联的问题也是很必要的。

12.3 通用数字信号处理器

通用数字信号处理器的基础是高速微处理器，为DSP操作优化了硬件体系结构和指令集。这些处理器广泛使用了并行处理、哈佛体系结构、流水线技术和专用硬件，专用硬件在任何可能的情况下执行例如移位/缩放、乘法等耗时操作。

按照计算效率、易实现性、成本、功耗、体积和应用专用需求来衡量，通用DSP处理器在过去十年获得了显著的发展，这是不懈追求更好地执行DSP操作方法的结果（Levy, 1998, 1999; Berkeley Design Technology, 1999）。对于改善计算效率的永不满足，已经显著地减少了指令周期时间，更重要的是增加了硬件和软件体系结构的复杂性。现在DSP处理器通常都具有专用的片上算术硬件单元（例如用来支持快速乘法/累加操作）、大容量多端口片上存储器和用于有效执行DSP中内核计算的特殊指令。我们也已经看到了增加数据字长（例如用来保持信号质量）和增加并行性（为了增加一个周期内执行的指令数和每个指令执行的操作数）的趋势。因此，我们发现在新的通用DSP处理器中，增加使用的是多数据通道/算术以支持并行操作。基于SIMD（单指令，多数据）、VLIW（超长指令字）和超标量体系结构的DSP处理器被引入以支持高效的并行处理。在一些DSP处理器中，通过使用专门的片上协处理器来加速特定的DSP算法（比如FIR滤波和维特比解码），进一步增强了相关的性能。通信和数字音频技术的爆炸性增长，已经对DSP处理器的发展产生了很大影响，比如在嵌入式DSP处理器应用方面的增长。

在接下来的内容中，我们将简要描述几代定点和浮点DSP处理器的体系结构特征。

12.3.1 定点数字信号处理器

现在可用的定点DSP处理器在体系结构细节和提供的板上资源方面是不同的。表12.1是四个领先的半导体厂商的四代定点DSP处理器的关键特征的总结。DSP处理器划分为四代的部分依据是按照历史原因、体系结构特征和计算性能。

表12.1 德州仪器、摩托罗拉和模拟器件公司的通用定点DSP处理器的特征

代	定点 DSP	数据通道宽度 (位)	数据通道个数	数据字长 (位)	累加器字长 (位)	指令宽度 (位)	片上RAM大小(字)	指令高速缓存大小 (指令个数)	乘法器个数	性能指标*
1	TMS320C10	16	1	16	32	16	144		1	
2	TMS320C50	16	2	16	32	16	10 K		1	10 @ 50 MHz
	DSP56002	24	2	24	56	24	1 K		1	13 @
	DSP-2100	16	2	16	40	24	32 K	16	1	13 @ 52 MHz
	1600	16	2	16	36	16		15	1	22 @ 120 MHz
3	TMS320C54	16	3	16	40	16	32 K		1	25 @ 100 MHz
	DSP56300	24	3	24	56	24		3 K	1	25 @ 100 MHz
	16000	32	2	32	40	32	127 K	31	2	36 @ 100 MHz
4	TMS320C6200		2		40	256	17 K	64 K	2	86 @ 133.6 MHz

*性能指标依据基准DSP核/算法的执行速度（Levy，1998；Berkeley Design Technology, 1999）。

由德州仪器在1982年首次发布的第一代定点DSP处理器系列（TMS320C1x）的基本体系结构如图12.15所示。TMS320C1x的关键特征是专用的算术单元，其包含一个乘法器和一个累加器。该处理器系列具有改进的哈佛体系结构，带有分离的程序和数据存储空间。它有一个片上存储器和用于执行基本DSP算法的特殊指令，尽管这些指令很有限。

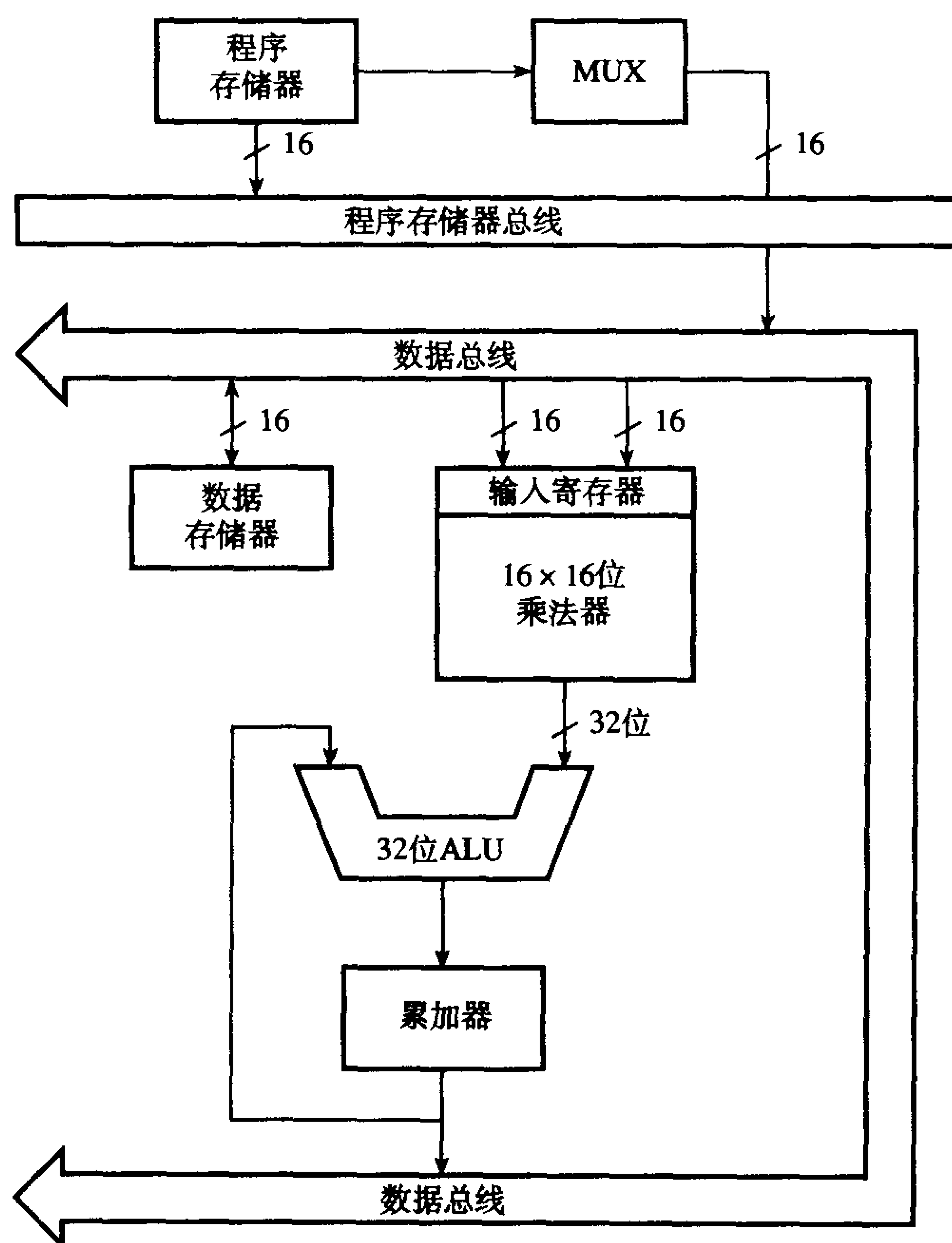


图12.15　一款第一代定点DSP处理器（德州仪器TMS320C10）的简化体系结构

第二代定点DSP处理器的性能和第一代相比有显著增强。在大多数情况下，这些增强的特征包括更大的片上存储器和更多的支持DSP算法有效执行的特殊指令。结果，第二代DSP处理器的计算性能是第一代的4～6倍。

典型的第二代DSP处理器包括德州仪器的TMS320C5x、摩托罗拉的DSP5600x、模拟器件公司的ADSP21xx和朗讯科技的DSP16xx系列。德州仪器的第一代和第二代DSP处理器在体系结构上有很多相似之处，但是第二代DSP处理器有更多的特征和更快的速度（参见表12.1）。TMS320C5x系列处理器的内部体系结构如图12.16所示，用一种简单的形式强调了双内部存储空间，这是哈佛体系结构的特征。DSP操作的特殊指令包括一个带数据移动的乘法和累加指令，这个指令可以和一个重复指令组合以执行一个FIR滤波器，能够显著地节省时间。它的倒位寻址性能对于FFT非常有用。不同于第一代定点处理器系列，C1x只有非常有限的内部存储器，C5x则提供了更多的片上存储器。

摩托罗拉的DSP5600x处理器是一款高精度的定点数字信号处理器。它的体系结构如图12.17所示。在内部，它有两个独立的数据存储空间——X数据和Y数据存储空间以及一个程序存储空

间。具有两个独立的数据存储空间允许DSP操作数据的自然分区，有利于算法的执行。例如在图形应用中数据可以存储为X和Y数据，在FIR滤波中作为系数和数据，在FFT中作为实部和虚部。在程序执行时，成对的数据样本可以在一个周期内同时提取或存储在内部存储器中。在外部，这两个数据空间复用单条数据总线，相应地会减少一些双内部数据存储器的好处。算术单元由两个56位的累加器和一个单周期定点硬件乘法－累加器（MAC）组成。MAC接受24位的输入，产生一个56位的乘积。24位字长为大部分DSP变量的表示提供了足够的精度，同时56位的累加器（包括8个保护位）防止算术运算溢出。这些字长对大多数应用是足够的，包括数字音频，其要求非常苛刻。5600x处理器提供特殊的指令，允许零开销的循环和倒位寻址能力，用于在FFT之前搅乱输入数据，或归整快速傅里叶变换后的数据。

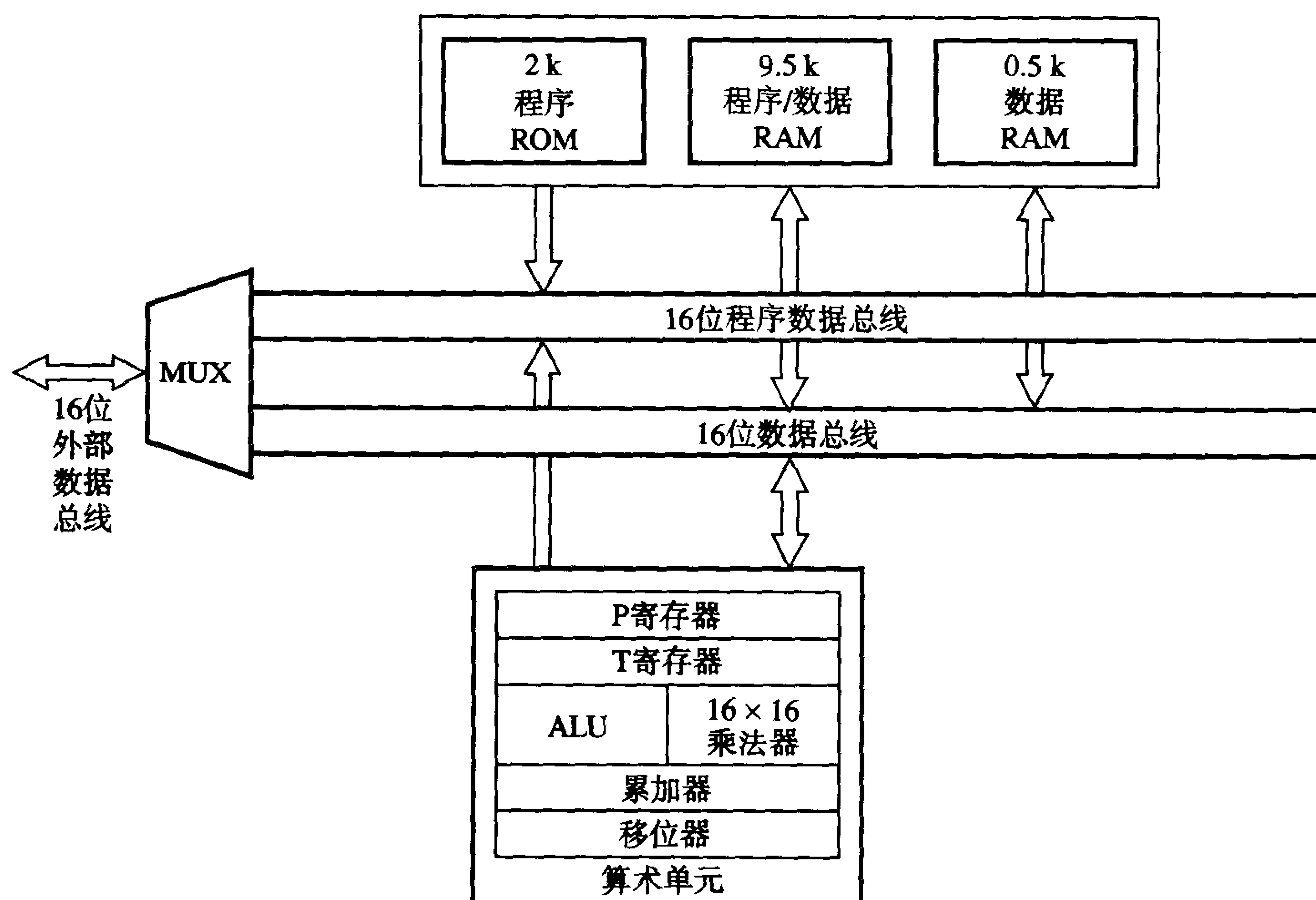


图12.16　一款第二代定点DSP处理器（德州仪器TMS320C50）的一种简化体系结构

模拟器件公司的ADSP21xx是第二代定点DSP处理器的另一个系列，它带有两个分离的外部存储空间，一个只存储数据，另一个不仅存储数据还存储程序代码。ADSP21xx内部体系结构的简化框图如图12.18所示。主要组件是ALU、乘法－累加器以及移位器。在一个周期内MAC接受16 × 16位的输入，产生一个32位的乘积。ADSP21xx的累加器有8个保护位，可以用于扩展精度。ADSP21xx不同于严格的哈佛体系结构，因为它允许在程序存储器中存储数据和程序指令。当数据和非程序指令从程序存储器中提取时，一条信号线（数据存取信号）用于给出指示。在程序存储器中，存储数据抑制了通过CPU的稳定的数据流，因为数据和指令提取不能同时发生。为了避免瓶颈，ADSP21xx系列有一个片上程序存储器高速缓存，用于保存最新执行过的16条指令。这消除了（特别是当执行程序循环时）从程序存储器中重复提取指令的需要。ADSP21xx提供特殊指令用于零开销循环，支持一种倒位寻址能力用于FFT。该处理器系列有一个大容量片上存储器（多达64 K字节的RAM用于增加数据传输性能）。处理器对DMA有出色的支持。外部器件可以与DSP处理器RAM传输数据和指令而不需要处理器干涉。

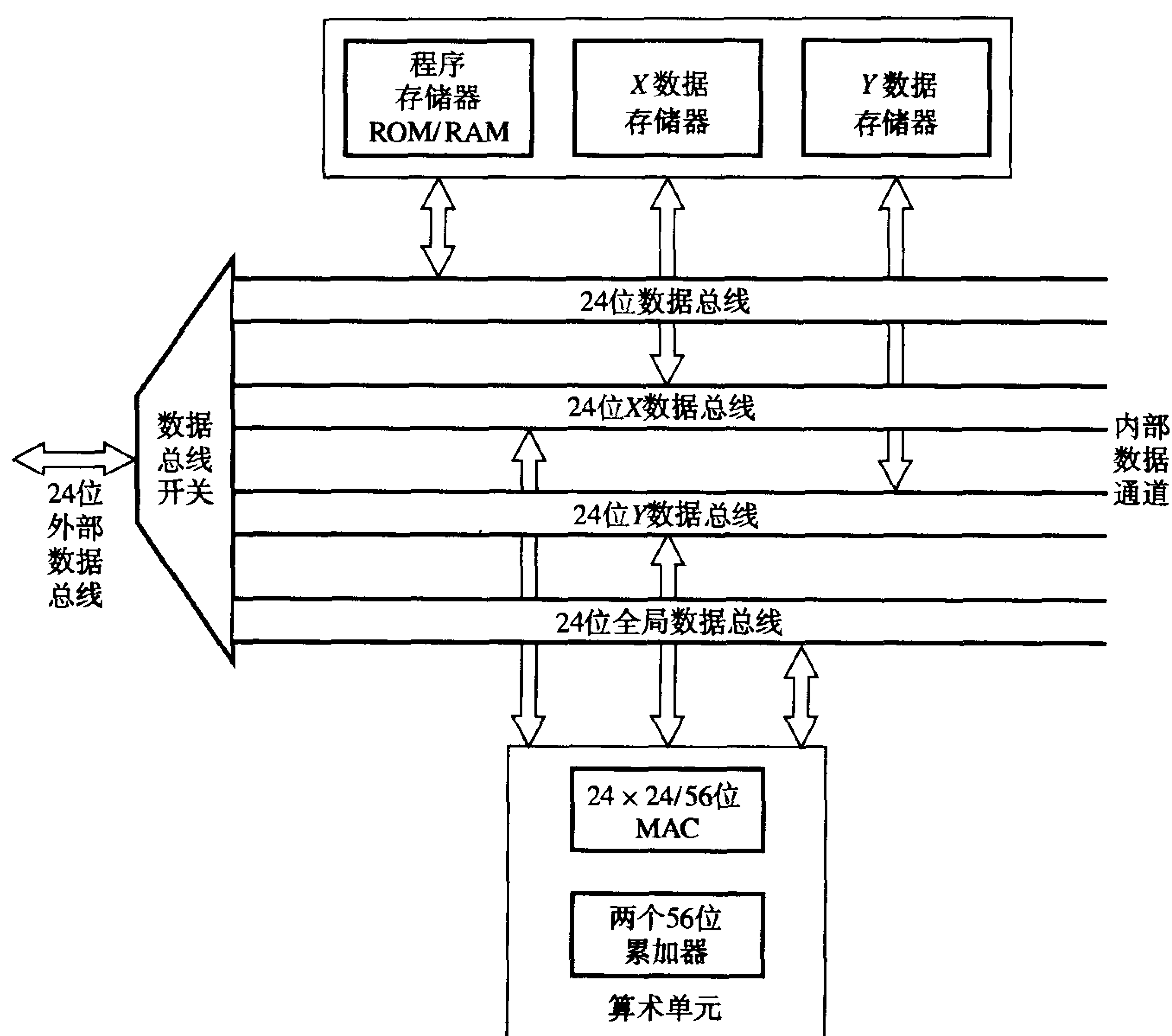


图 12.17　一款第二代定点 DSP 处理器（摩托罗拉 DSP56002）的简化体系结构

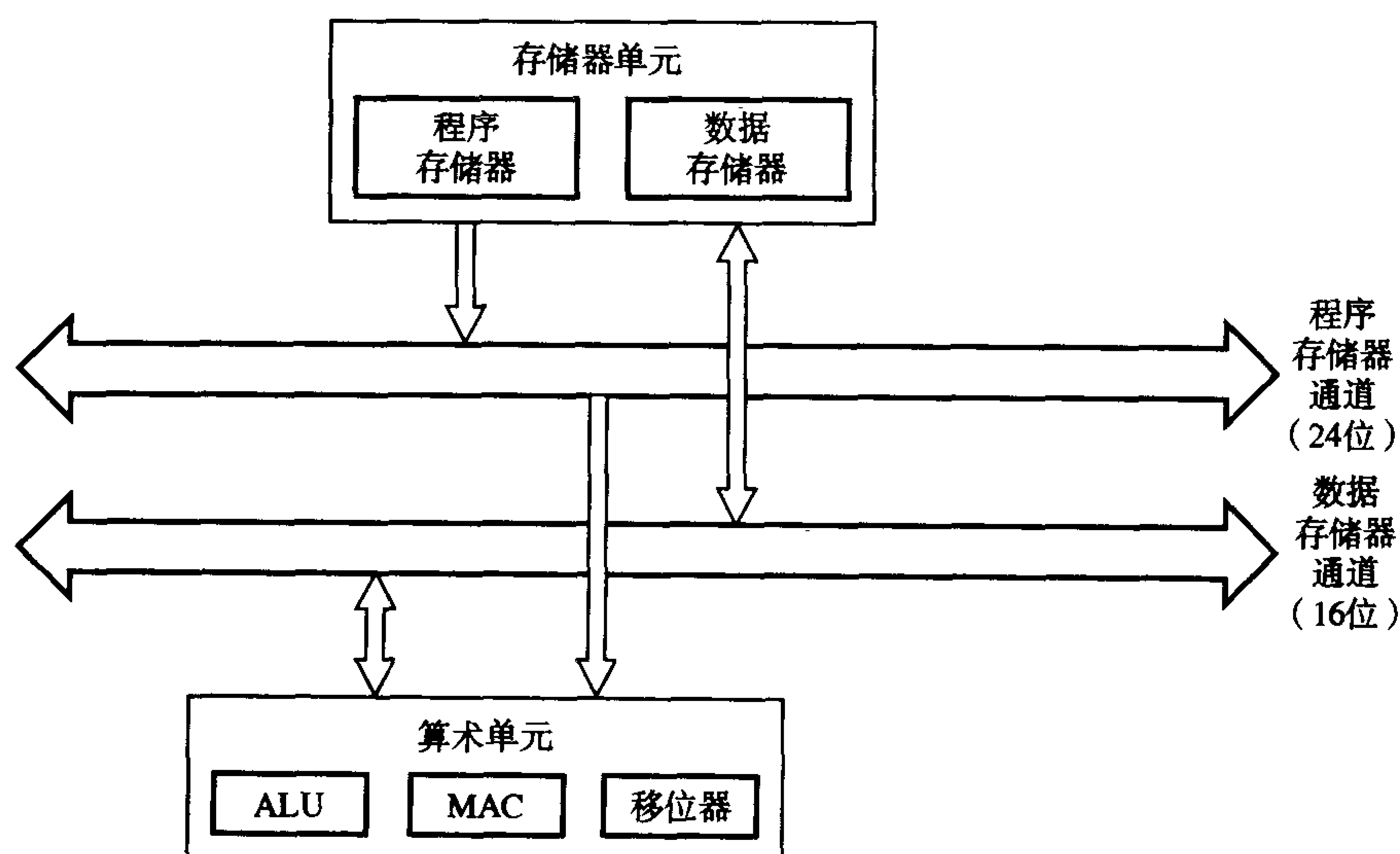


图 12.18　一款第二代定点 DSP 处理器（模拟器件公司 ADSP2100）的简化体系结构

朗讯科技的 DSP16xx 系列定点 DSP 处理器（参见图 12.19）瞄准的是通信和调制解调器市场。根据计算性能，它是最强大的第二代处理器之一。该处理器采用哈佛体系结构，和大多数其他第二代处理器相似，它有两条数据通道：X 和 Y 数据通道。它的数据算术单元包括一个专用的 16 × 16 位的乘法器，一个 36 位 ALU/移位器（其中包括 4 个保护位）和双累加器。还提供了例如零开销的单指令和指令块循环等特殊指令。

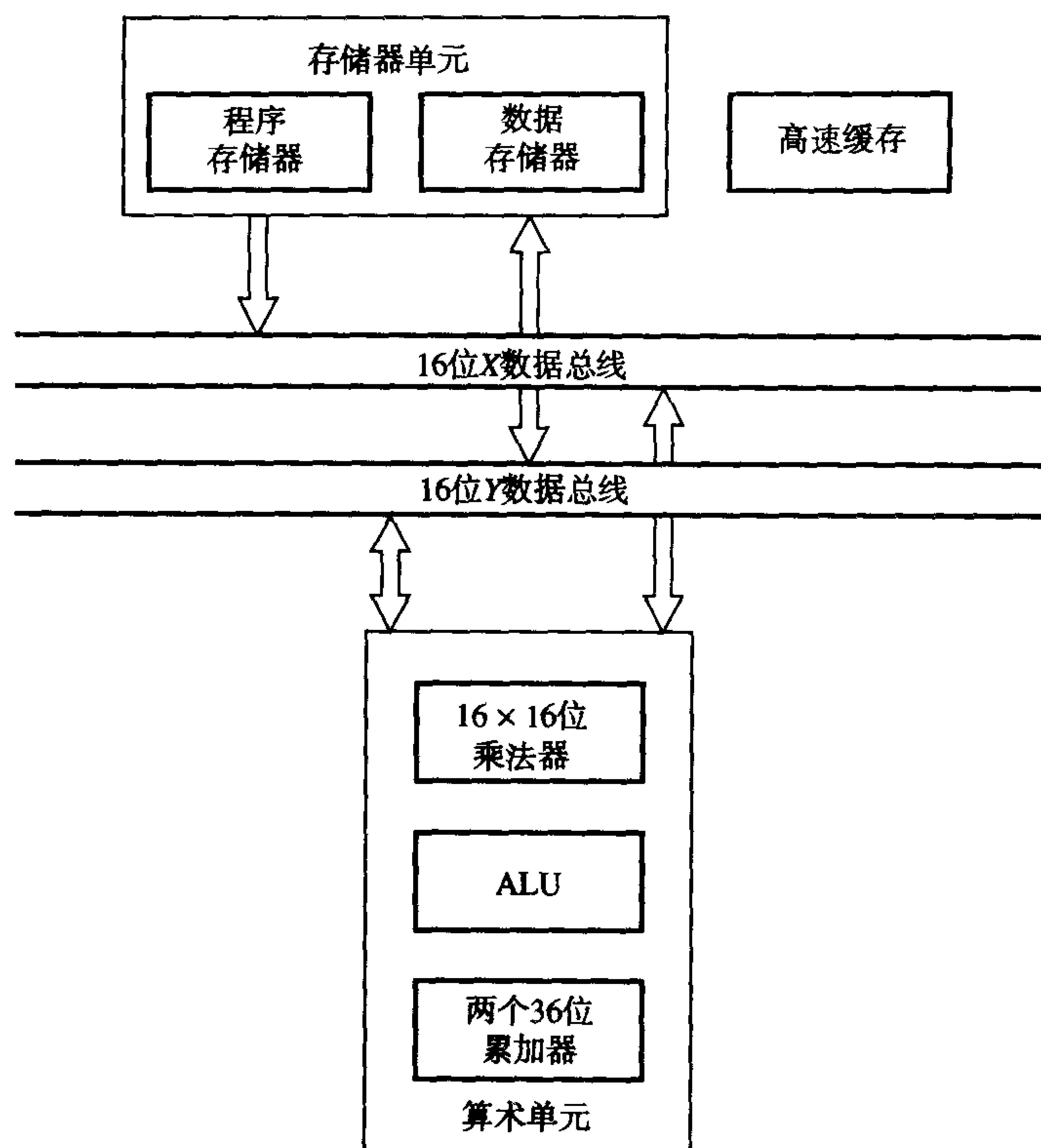


图 12.19　朗讯科技的DSP16xx定点DSP处理器的简化体系结构

第三代定点DSP处理器基本上是第二代DSP处理器的增强。总地来说，性能增强是通过增加和/或更有效地使用可使用的片上资源来实现的。和第二代DSP处理器相比，第三代DSP处理器的特征包括更多的数据通道（通常有3个，相比于第二代的2个），更宽的数据通道，更大的片上存储器和指令高速缓存，以及在某些DSP处理器中的一个双MAC。结果，第三代DSP处理器的性能通常要比相同系列的第二代DSP处理器好两到三倍(Levy, 1998; Berkeley Design Technology, 1999)。三款第三代DSP处理器TMS320C54x、DSP563x和DSP16000的简化体系结构如图12.20、图12.21和图12.22所示。大多数第三代定点DSP处理器瞄准的应用是数字通信和数字音频，反映了这些领域的巨大增长和对DSP处理器发展的影响。因此，我们在一些处理器中发现了支持这些应用的特征。例如，TMS320C54x包含特殊指令用于自适应滤波（经常用于电信中的回声消除和自适应均衡）和支持维特比解码。在第三代处理器中，半导体厂商对于功耗问题也非常认真（因为在便携式和手持式设备比如手机中的重要性）。大多数第三代DSP处理器是低功耗的，具有电源管理功能。

新体系结构的第四代定点DSP处理器的主要目标是大的和/或新兴的多通道应用，比如数字用户环（digital subscriber loop）、远程访问服务器调制解调器、无线基站、第三代移动系统和医疗成像。在DSP社区吸引了极大注意力的新型定点体系结构是超长指令字（VLIW）（更多细节请参见12.2节）。新体系结构在保持上一代DSP处理器的一些好特征的同时，充分使用了并行技术。和上一代相比，第四代定点DSP处理器总地来说有更宽的指令字、更宽的数据通道、更多寄存器、更大的指令高速缓存和多算术单元，使它们能够在每个周期执行更多的指令和操作。

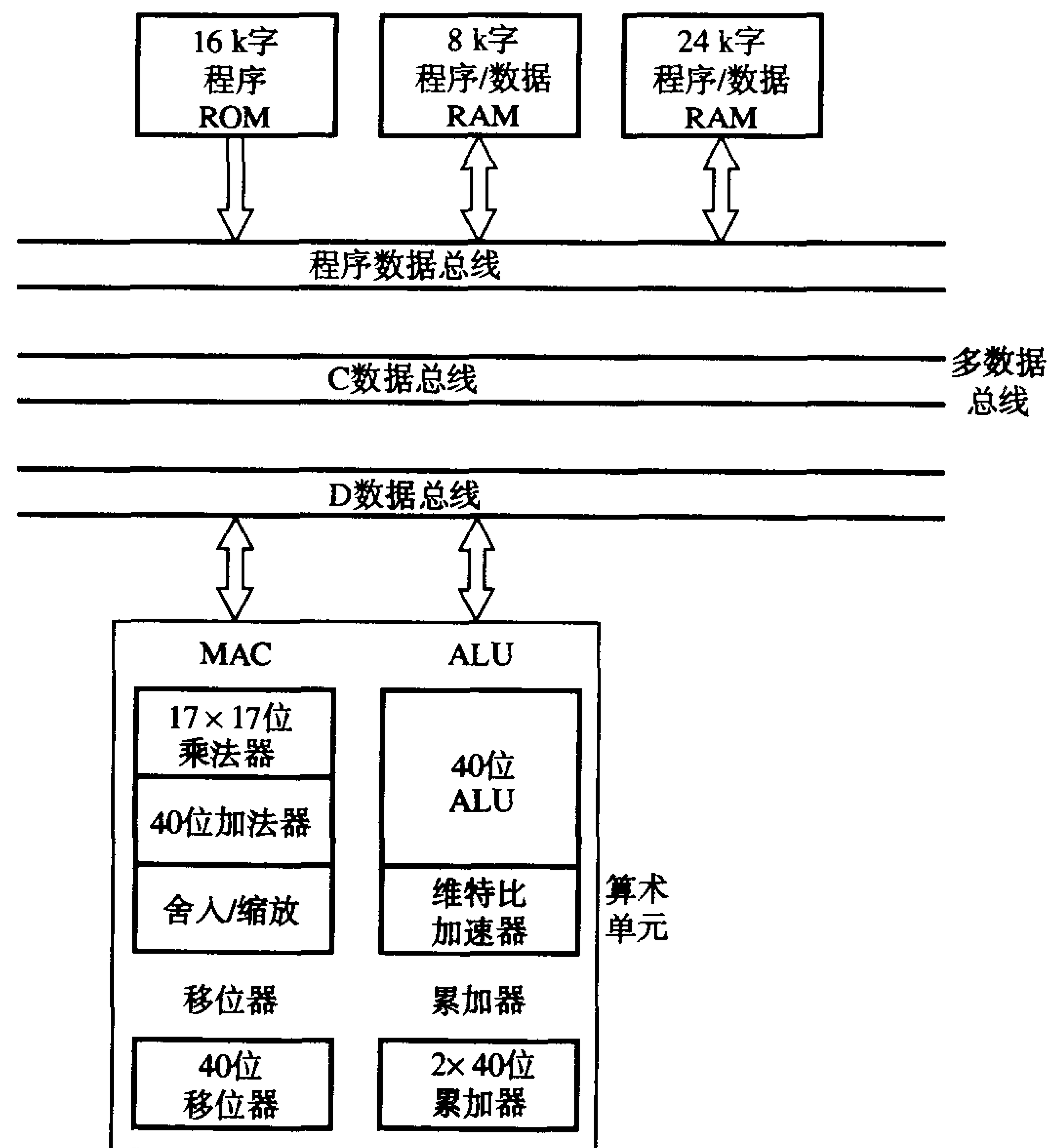


图 12.20　一款第三代定点 DSP 处理器（德州仪器 TMS320C54x）的简化体系结构

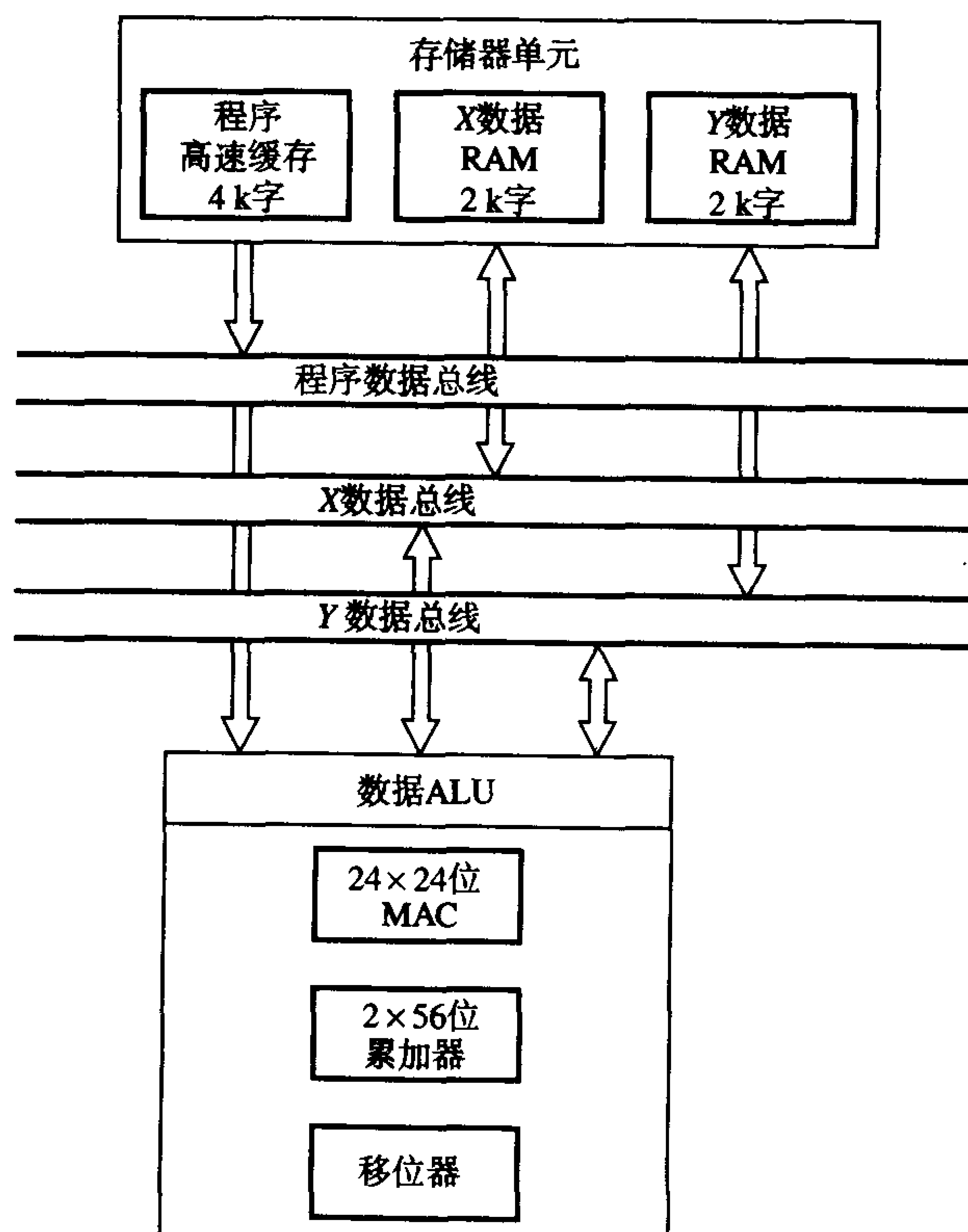


图 12.21　一款第三代定点 DSP 处理器（摩托罗拉 DSP56300）的简化体系结构

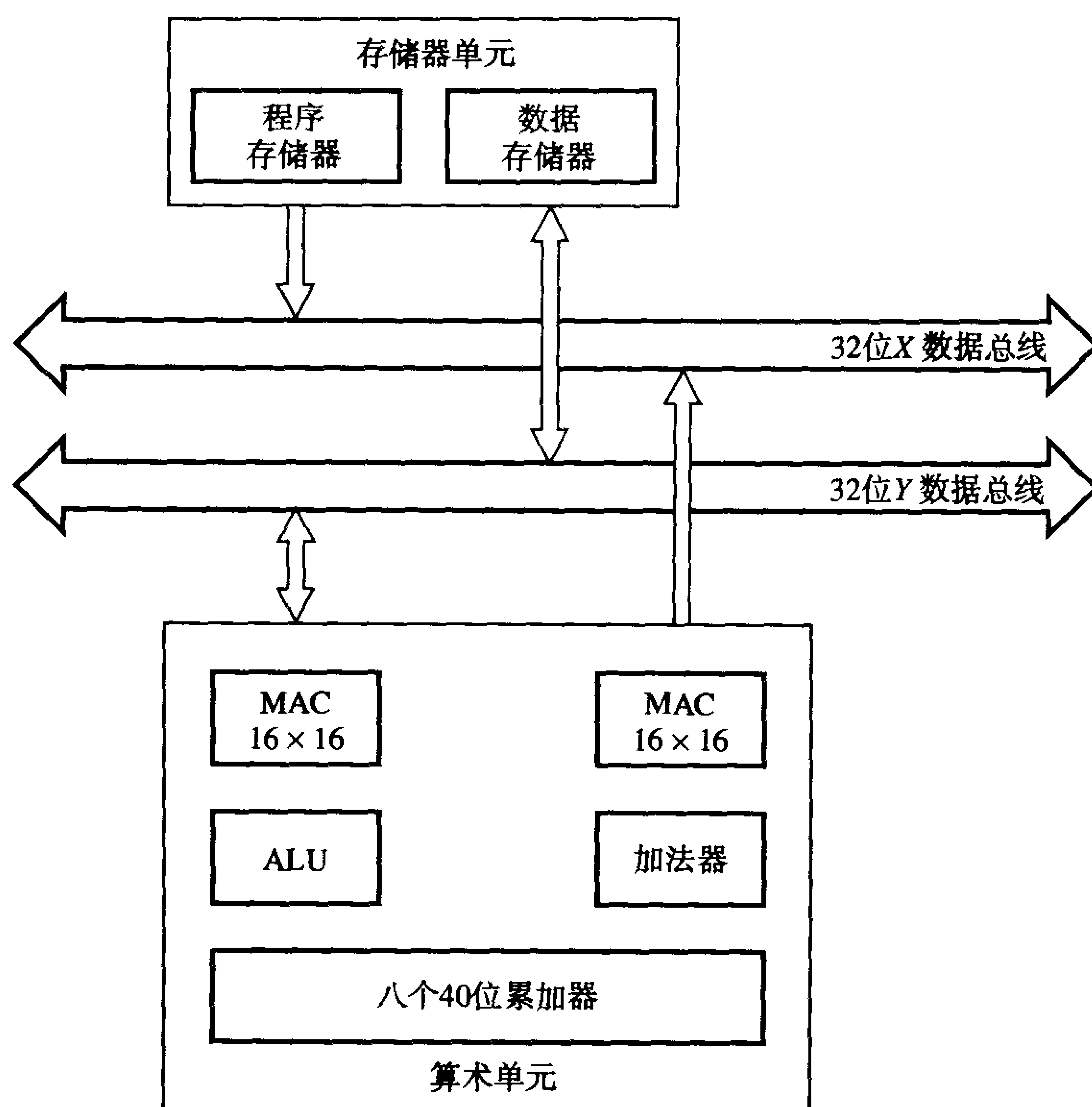


图12.22 一款第三代定点DSP处理器（朗讯科技DSP16000）的简化体系结构

德州仪器的TMS320C62x系列定点DSP处理器是基于VLIW体系结构的，请参见图12.23。其核心处理器有两个独立的算术通道，每个有4个执行单元：一个逻辑单元（Li），一个移位器/逻辑单元（Si），一个乘法器（Mi），以及一个数据寻址单元（Di）。通常，核心处理器每次提取8个32位的指令，并给出一个256位宽度的指令（这就是术语超长指令字）。利用全部8个执行单元，每个数据通道4个，TMS320C62x能够在一个周期内并行执行多达8条指令。处理器有一个大的程序和数据高速缓存存储器（通常，4 K字节的一级程序/数据高速缓存和64 K字节的二级程序/数据高速缓存）。每个数据通道有自己的寄存器文件（16个32位的寄存器），但是也可以访问其他数据通道的寄存器。VLIW体系结构的优点包括简单性和高计算性能。缺点包括增加了程序存储器的使用（与处理器内在并行性相匹配的代码的组织可能会使存储器的使用效率较低）。进一步，最佳的处理器性能只有当所有的执行单元都忙时才能获得，这并不总是可能的，因为数据的依赖性、指令的延迟和执行单元使用的限制。然而，用于代码打包、指令规划、资源分配，总体上说用来开发处理器巨大潜能的复杂的编程工具是可以得到的。

12.3.2 浮点数字信号处理器

DSP处理器使用浮点算术执行高速、高精度DSP操作的能力已经有了可喜的发展。这使DSP固有的有限字长效应比如溢出、舍入误差以及系数量化误差最小化。它也有利于算法的开发，因为设计者可以在大型机上使用高级语言开发算法，然后移植到DSP器件，这要比使用定点处理器更容易。

浮点DSP处理器保留了定点处理器的关键特征，比如用于DSP操作的特殊指令和用于多操作的多数据通道。和定点DSP处理器一样，可用的浮点DSP处理器的体系结构显著不同。德州仪器和模拟器件公司的第三代浮点DSP处理器的一些关键特征总结在表12.2中。

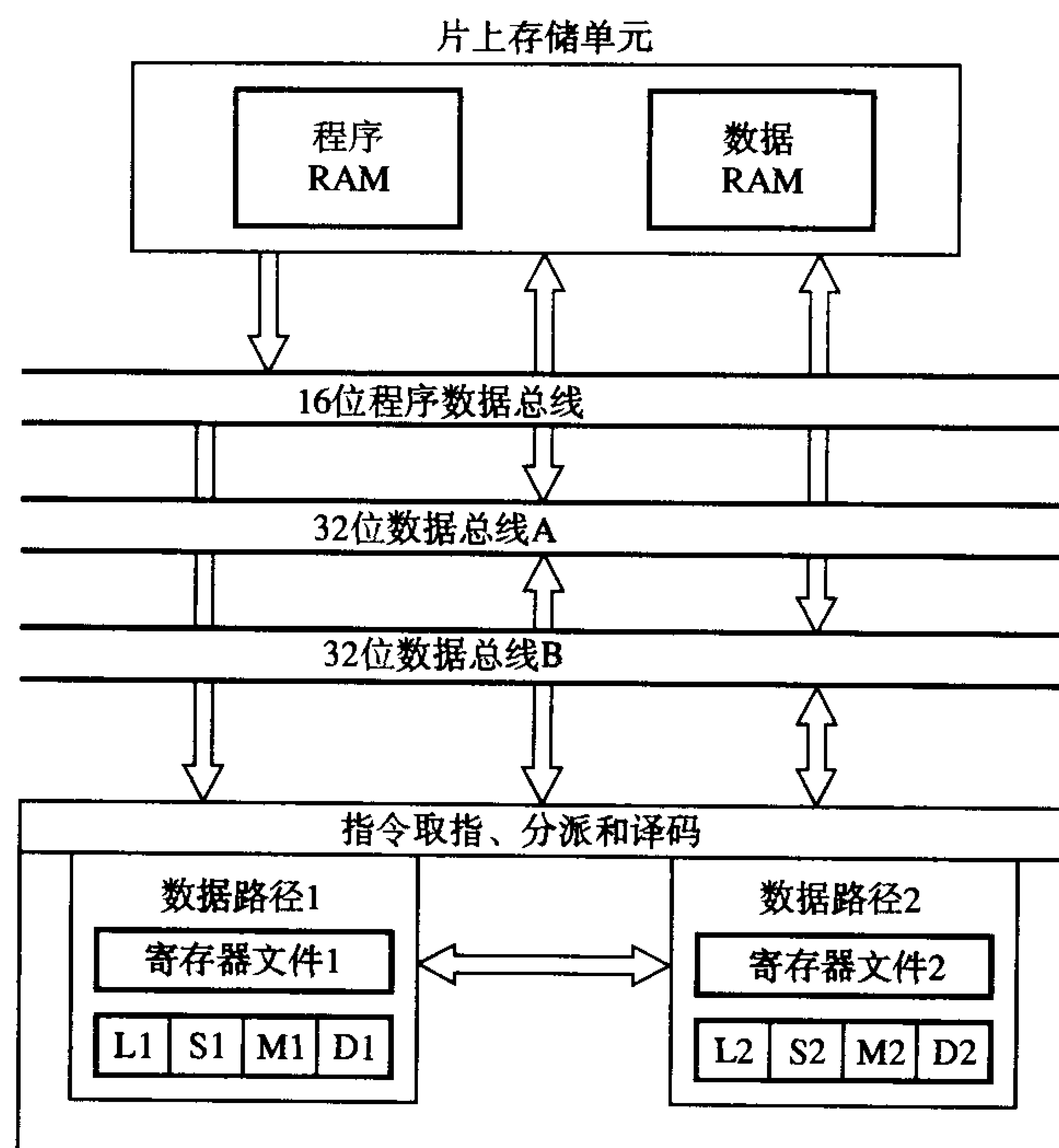


图 12.23　一款第四代定点、长指令DSP处理器（德州仪器TMS320C62x）的简化体系结构。注意两个独立的算法数据路径，每一个带有 4 个执行单元：L1、S1、M1 和 D1（L2、S2、M2 和 D2）

表 12.2　德州仪器和模拟器件公司的通用浮点 DSP 处理器的特征

代	浮点 DSP 处理器	数据通道宽度 (位)	数据通道个数	数据字长 (位)	累加器字长 (位)	指令宽度 (位)	片上 RAM 大小(字)	指令高速缓存大小 (指令个数)	乘法器个数	性能指标 *
1	TMS320C30	16	1	32	40	32	2 K	64	1	7 @ 30 MHz
2	TMS320C40	16	2	32	40	32	2 K	128	1	7 @ 30 MHz
	ADSP-21060	24	2	32	80	48	128 K	32	1	14 @ 50 MHz
3	TMS320C67x	16	3	16	40		17 K		2	
	TigerSHARC	128	3	32	40/80	128	192 K		2	

* 性能指标依据基准 DSP 核 / 算法的执行速度（Levy，1998；Berkeley Design Technology，1999）。

TMS320C3x大概是第一代通用浮点DSP处理器中最知名的系列。C3x系列是32位单片数字信号处理器，支持整数和浮点算术操作。它们有很大的存储空间，配备了很多片上外围设备以简化系统设计。这些设备包括一个程序高速缓存以改善常用代码的执行，以及片上双端口存储器。大存储空间适用于存储密集型应用，例如图形和图像处理。在TMS320C30中，一个乘法要求32位的操作数并且产生一个40位的归一化的浮点乘积。整数乘法要求24位的输入并且产生32位的结果。C3x系列支持三种浮点格式。第一种是 16 位短浮点格式，即 4 位指数、1 位符号位和 11 位尾数（mantissa）。这种格式用于立即（immediate）浮点操作。第二种是单精度浮点格式，即 8 位指数、1 位符号位和 23 位小数（总共 32 位）。第三种是 40 位扩展精度格式，即 8 位指数、1 位符号位和 31 位小数。这种浮点表示和标准的IEEE格式不同，但是提供允许在两种格式之间转换的工具。TMS320C3x组合了哈佛体系结构（分离的程序指令、数据和I/O总线）和冯·诺伊曼处理器（统一的寻址空间）的特征。

第二代通用浮点DSP处理器的重点在于多处理和多处理器的支持。多处理器支持的关键问题包括处理器间的通信、DMA传输和全局存储器共享。最有名的第二代浮点DSP处理器系列是德州仪器的TMS320C4x和模拟器件公司的ADSP-2106x SHARC（超级哈佛体系结构计算机）。C4x共享了C3x的一些体系结构特征，但是它是为多处理而设计的。C40x系列有很好的I/O能力——它有6个COMM端口用于处理器间的通信，以及6个32位宽的DMA通道用于快速数据传输。这种体系结构允许在一个指令周期内并行执行多个操作。C4x系列支持浮点和定点算术。C40的本地浮点数据格式不同于IEEE的754/854标准，尽管它们之间的转换很容易实现。

模拟器件公司的ADSP-2106x SHARC DSP处理器也是32位浮点器件。它们有大的内部存储器和出色的I/O能力——10个DMA通道允许无干涉地访问内部存储器，以及6个Link端口用于处理器之间的高速通信。这种体系结构允许共享全局存储器，可以使多达6个SHARC处理器以全数据速率相互访问内部RAM。ADSP-2106x系列支持定点和浮点算术。它的单精度浮点格式和单精度IEEE 754/854浮点标准（24位尾数和8位指数）是兼容的。这种体系结构也支持每个周期的多操作。

第三代浮点DSP处理器采用了更多的并行处理技术，以增加一个周期内的指令数和操作数，从而迎接多通道和计算密集型应用的挑战。这是通过使用新型体系结构实现的，特别是VLIW（超长指令字）和超标量体系结构。两种领先的第三代浮点DSP处理器系列是德州仪器的TMS320C67x和模拟器件公司的ADSP-TS001。TMS320C67x系列和先进的第四代定点DSP处理器TMS320C62x（更多细节参见图12.23和12.3.1节）具有相同的VLIW体系结构。

TigerSHARC DSP处理器系列支持混合的算术类型（定点和浮点算术）和数据类型（8位、16位和32位数）。这种灵活性使得针对给定的应用采取最合适的算术和数据类型以增强性能成为可能。和TMS320C67x一样，TigerSHARC也是瞄准大规模、多通道应用，比如第三代移动系统（3G无线）、数字用户线（xDSL）和用于Internet服务的远程多访问服务器调制解调器。静态超标量体系结构的TigerSHARC（更多细节请参见图12.14和12.3.1节）组合了VLIW体系结构、常规DSP体系结构和RISC计算机的优良特征。该处理器有两个计算单元，每个带有一个乘法器、ALU和64位移位器。该处理器每个周期能够执行多达八个16位输入和40位累加的MAC操作、两个16位复数数据的40位MAC操作或两个32位数据的80位MAC操作。使用8位数据，TigerSHARC在一个周期内能够执行多达16个操作。TigerSHARC具有很宽的存储器带宽，它的存储器组织在3个128位宽的单元中。访问数据可以使用可变的数据大小——正常的32位字、长64位字或4个128位字。在一个周期内能够提交多达4个32位的指令。为了避免使用大量的NOP（这是VLIW设计的一个缺点），大的指令字可以分解为分离的短指令，从而独立地分配给每个单元。

12.4　选择数字信号处理器

针对给定的应用选择合适的DSP处理器近年来已经成为一个重要的主题，因为可供选择的处理器的范围很宽（Levy, 1999; Berkeley Design Technology, 1996, 1999）。为应用选择DSP处理器时可能需要考虑的特殊因素包括体系结构特征、执行速度、算术类型和字长。

(1) **体系结构特征**　现在可用的大多数DSP处理器都有好的体系结构特征，但是不一定满足特定应用的要求。感兴趣的关键特征包括片上存储器的大小、特殊指令和I/O性能。在大多数实时DSP应用中，为了迅速地访问数据和快速地执行程序，片上存储器是基本需求。对于需要大量存储器的应用（例如数字音频——Dolby AC-2，FAX/Modem，MPEG编/解

码），内部RAM的大小可能成为一个重要的区分因子。如果内部存储器不足，可以通过高速的片外存储器而相应地增加，尽管这可能会增加系统成本。对于要求和外部世界进行快速有效的通信或数据传输的应用，I/O性能（比如ADC和DAC的接口）、DMA性能和对多处理的支持可能会非常重要。依赖于应用，支持DSP操作的丰富的特殊指令集是非常重要的，例如零开销的循环能力、专用的DSP指令和循环寻址。

(2) **执行速度**　DSP处理器的速度是性能的一个重要衡量指标，这是因为大多数DSP任务的时间紧要性（time-critical）的本质。传统上，执行速度的两个主要衡量单位是处理器的时钟速度，用MHz表示；以及执行的指令数，用每秒百万条指令（MIPS）或者在浮点DSP处理器中用每秒百万浮点操作数（MFLOPS）表示。然而，这两种衡量指标在某些情况下可能是不合适的，因为不同DSP处理器操作方式的显著不同，大多数都可以在一个机器指令内执行多个操作。例如，C62x系列处理器在一个周期内能够执行八条指令。在每个周期中执行的操作数也是随着处理器的不同而不同。因此，基于这些指标的处理器执行速度的比较可能是没有意义的。基于基准算法——例如DSP核（比如FFT、FIR和IIR滤波器）的执行速度是一个可供选择的衡量方法（Levy, 1998; Berkeley Design Technology, 1999）。在表12.1和表12.2中，基于这些基准的性能指标给出了许多流行DSP处理器相对性能的一个指示。

(3) **算术类型**　在现代DSP处理器中用到的两种最常见的算术类型是定点和浮点算术。浮点算术是具有很宽和可变动态范围（动态范围可以定义为最大和最小信号电平之间的差值，或者最大信号和噪声基底的差值，以分贝表示）要求的应用的自然选择。定点处理器对于低成本、大批量应用（例如蜂窝电话和计算机磁盘驱动）很有好处。使用定点算术会产生和动态范围约束相关的问题，这是设计者必须解决的（更多细节请参见第13章）。一般来说，浮点处理器比定点处理器更昂贵，尽管差价近年来已经显著降低。大多数现在可用的浮点DSP处理器也支持定点算术。

(4) **字长**　处理器字长是DSP中的一个重要参数，因为它对信号指令有显著影响。它决定了DSP操作的参数和结果能够表示得如何准确（更多细节请参见第13章）。总地来说，数据字越长，数字信号处理引入的误差就越小。例如在定点音频处理中，要求至少24位处理器字长以使最小信号电平显著地高于信号处理产生的噪声基底，以保证CD质量。在定点DSP处理器中，根据应用，使用可变的处理器字长（参见表12.1）。瞄准电信市场的定点DSP处理器趋向于使用16位字长（例如TMS320C54x），而瞄准高质量音频应用的处理器趋向于使用24位字长（例如DSP56300）。近些年来，我们已经看到因为器件的成本下降而使用更多位数的ADC和DAC（例如Cirrus的24位音频编解码器，CS4228），以满足增加质量的无尽需求的趋势。因此，我们可能会看到对音频处理增加更大的处理器字长的需求。在定点处理器中，也可能有必要在累加器中提供保护位（通常1～8位），以防止在扩展乘法和累加操作时的算术溢出。DSP处理器中的附加位有效地扩展了可用的动态范围。在大多数浮点DSP处理器中，单精度算术使用32位数据大小（24位尾数和8位指数）。这个大小也是和IEEE浮点格式（IEEE 754）相兼容的。大多数浮点DSP处理器也具有定点算术能力，经常支持可变数据大小的定点算术。

实际上，一些因素比如对特定DSP处理器系列的经验/熟悉程度、易用性、面市时间和成本，可能是选择特定处理器时最需要考虑的。

12.5 DSP 算法在通用数字信号处理器上的实现

12.5.1 FIR 数字滤波

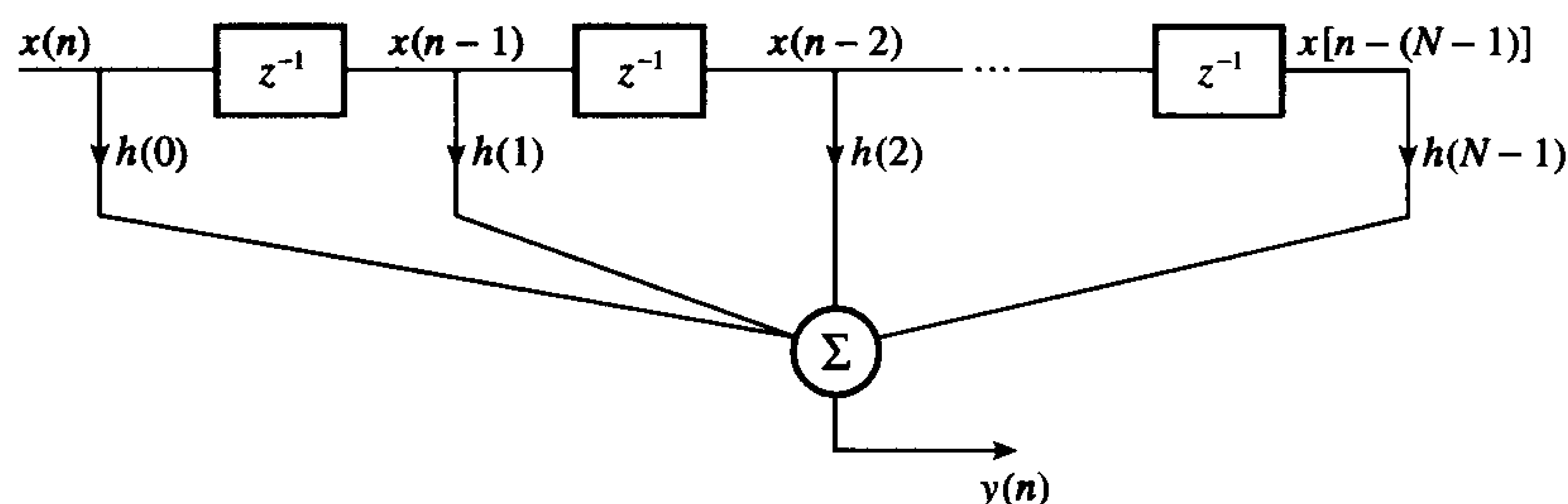
具有图 12.24(a)结构的、非递归的 N 点 FIR 滤波器由下面的差分方程（细节请参见第 7 章）表征：

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k) \tag{12.4}$$

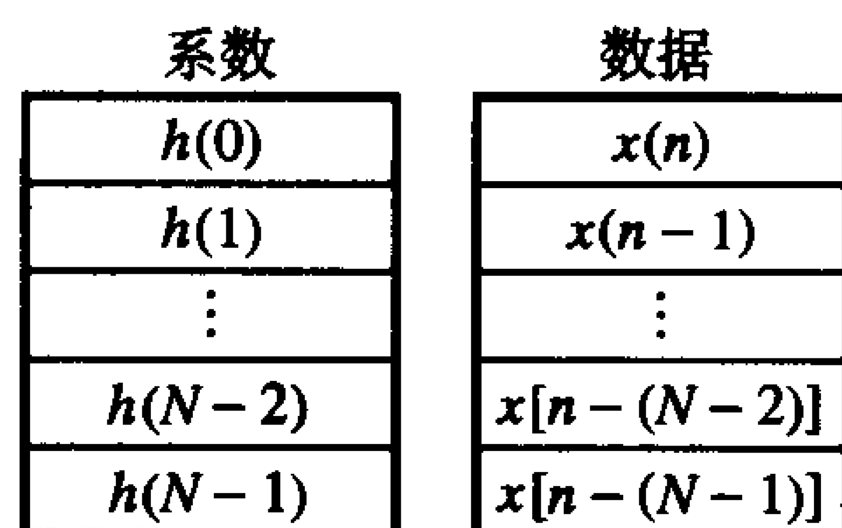
程序 12.1 给出了用 C 语言实现的通用 FIR 滤波器的一个片断。对于实时 FIR 滤波，数据和系数存储在存储器中，图 12.24(b)是概念性的示意图。为了理解 FIR 滤波器是如何工作的，考虑 $N=3$ 的简单情况，使用下面的差分方程：

$$y(n) = h(0)x(n) + h(1)x(n-1) + h(2)x(n-2) \tag{12.5}$$

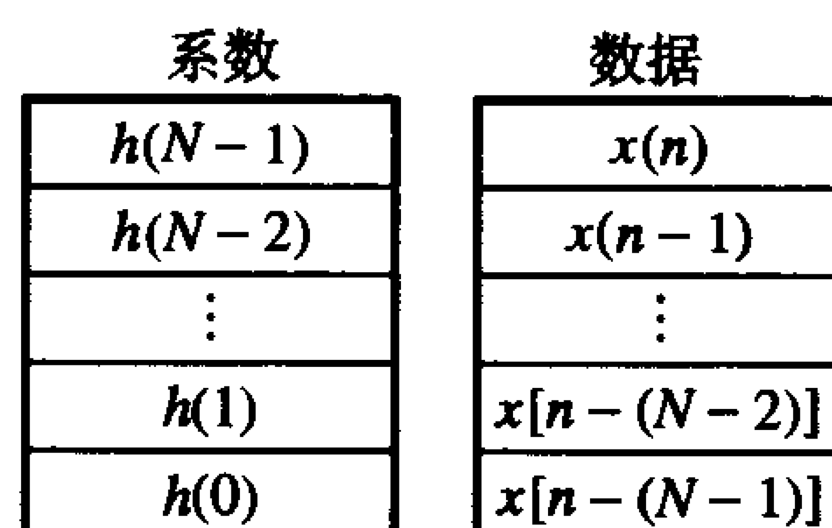
$x(n)$是最新的输入样本，$x(n-1)$是上一个样本，$x(n-2)$是再之前的一个样本。



(a) 滤波器结构



(b) 系数和数据存储器映像



(c) 可选择的存储器映像

图 12.24 FIR 滤波器的实现

程序 12.1 FIR 滤波的 C 语言伪代码

```
nm1=N-1;
yn=0;
for(k=0; k<nm1;++k){          /* shift data to make room for new sample */
      x[nm1-k]=x[nm1-k-1];
      x[0]=xn;
}
for(k=0; k<N;++k){
      yn=yn+h[k]*x[k];        /* filter data and compute output sample */
}
return(yn);                   /* filter output sample */
```

假定这个3系数的数字滤波器从一个ADC获得输入。首先要做的事情是分配两组连续的存储位置（在RAM中），一组用于存储输入数据（$x(n)$，$x(n-1)$，$x(n-2)$），另一组用于滤波器系数（$h(0)$，$h(1)$，$h(2)$），如下所示：

数据 RAM	系数 存储器
0	$h(0)$
0	$h(1)$
0	$h(2)$

在初始化时，存储数据样本的RAM位置设置为零，因为我们总是从没有数据时开始。然后执行下列操作。

(1) **第一个抽样时刻**　从ADC读取数据样本，将数据RAM移动一个位置（为新数据腾出空间），保存新输入数据，根据12.5式计算输出样本然后将计算出的输出样本送给DAC：

数据 RAM	系数 存储器	
→$x(1)$	$h(0)$	$y(1) = h(0)x(1) + h(1)x(0) + h(2)x(-1)$
0	$h(1)$	
0	$h(2)$	

(2) **第二个抽样时刻**　重复以上操作，计算出新的输出样本并送给DAC：

数据 RAM	系数 存储器	
→$x(2)$	$h(0)$	$y(2) = h(0)x(2) + h(1)x(1) + h(2)x(0)$
$x(1)$	$h(1)$	
0	$h(2)$	

(3) **第三个抽样时刻**　重复以上操作，计算出新的输出样本并送给DAC：

数据 RAM	系数 存储器	
→$x(3)$	$h(0)$	$y(3) = h(0)x(3) + h(1)x(2) + h(2)x(1)$
$x(2)$	$h(1)$	
$x(1)$	$h(2)$	

(4) **第四个抽样时刻**　重复以上操作，计算出新的输出样本并送给DAC：

数据 RAM	系数 存储器	
→$x(4)$	$h(0)$	$y(4) = h(0)x(4) + h(1)x(3) + h(2)x(2)$
$x(3)$	$h(1)$	
$x(2)$	$h(2)$	

注意到最早的数据样本现在已经落在最后。

(5) **第 n 个抽样时刻** 重复以上操作，计算出新的输出样本并送给DAC：

数据 RAM	系数存储器	
→ $x(n)$	$h(0)$	$y(n)=h(0)x(n)+h(1)x(n-1)+h(2)x(n-2)$
$x(n-1)$	$h(1)$	
$x(n-2)$	$h(2)$	

用第一代定点DSP处理器（TMS320C10）实现的三点FIR滤波器在程序12.2中给出。在此例中，乘积的计算从数据和系数的底部开始，以利用TMS320C10的数据移动指令。指令LTD和MPY是基于TMS320C10实现的FIR滤波器的中心。例如，下面两个指令执行在12.4式中隐含的或者在图12.24(a)中用 z^{-1} 表示的移位，将前一个乘积加到累加器，计算新的乘积 $h(k)x(n-k)$。

```
LTD    XNM1
MPY    H1
```

特别是指令LTD XNM1将数据样本 $x(n-1)$（保存在数据RAM地址XNM1中）装载到T（临时）寄存器，将仍在P（乘积）寄存器中的前一个乘积 $h(2)x(n-2)$ 加到累加器，将 $x(n-1)$ 移到下一个地址，即 $x(n-2)=x(n-1)$。第二个指令MPY将T寄存器的内容和 $h(1)$ 相乘，结果留在乘积寄存器。移位策略保证当计算下一个样本的时候，输入样本在正确的位置。

程序12.2 三点FIR滤波器的简洁代码

```
NXTPT   IN      XN, ADC
        ZAC
        LT      XNM2
        MPY     H2        ;h(2)x(n-2)

        LTD     XNM1      ;0+h(2)x(n-2); x(n-2)=x(n-1)
        MPY     H1        ;h(1)x(n-1); x(n-1)=x(n-2)

        LTD     XN        ;h(2)x(n-2)+h(1)x(n-1); x(n-1)=x(n)
        MPY     H0        ;h(0)x(n)

        APAC              ;h(2)x(n-2)+h(1)x(n-1)+h(0)x(n)

        SACH    YN,1      ;save output sample
        OUT     YN,DAC    ;output sample to DAC

        B       NXTPT
```

FIR滤波器的简洁代码是一个快速实现，但不通用，因为大 N 点滤波器不会产生紧凑的代码。特别是，通用FIR滤波器是通过设置一个内循环执行FIR公式并且计算12.4式规定的滤波器输出而实现的。

显示出内循环的 N 点FIR滤波器的流程图在图12.25中给出。在第一代DSP滤波器中，FIR滤波器的内循环可以通过下列指令执行：

```
LOOP LTD     *, AR0    ; shift/update delay line and accumulate products
     MPY     *-, AR1   ; multiply next coefficient and data value
     BANZ    LOOP
```

在此例中，辅助寄存器AR0和AR1用于指向相乘的数据和系数。辅助寄存器AR1包含滤波器长度，作为循环计数器。寄存器的非零分支指令BANZ和AR1一起，用于控制循环。在第一代DSP处理器中实现的FIR滤波器不是很有效，这是因为循环控制的开销。

第二代定点DSP处理器，比如TMS320C50和摩托罗拉的DSP56000，具有零开销循环能力和特殊的乘法和累加指令，可以帮助削减执行FIR内循环的时间。在TMS320C50中，图12.25中显示的 N 点FIR滤波器的内循环可以使用下列指令而有效地执行：

```
RPT     NM1
MACD    HNM1, XNM1
```

指令RPT NM1将滤波器长度减1($N-1$)载入重复寄存器，使带有数据移位的乘法和累加指令MACD零开销地执行 $N-1$ 次。MACD将LTD和MPY两个指令组合成单个指令，这样就能够更快执行。指令 RPT 和 MACD 是 DSP 处理器中可用的、省时的特殊指令的一个好例子。

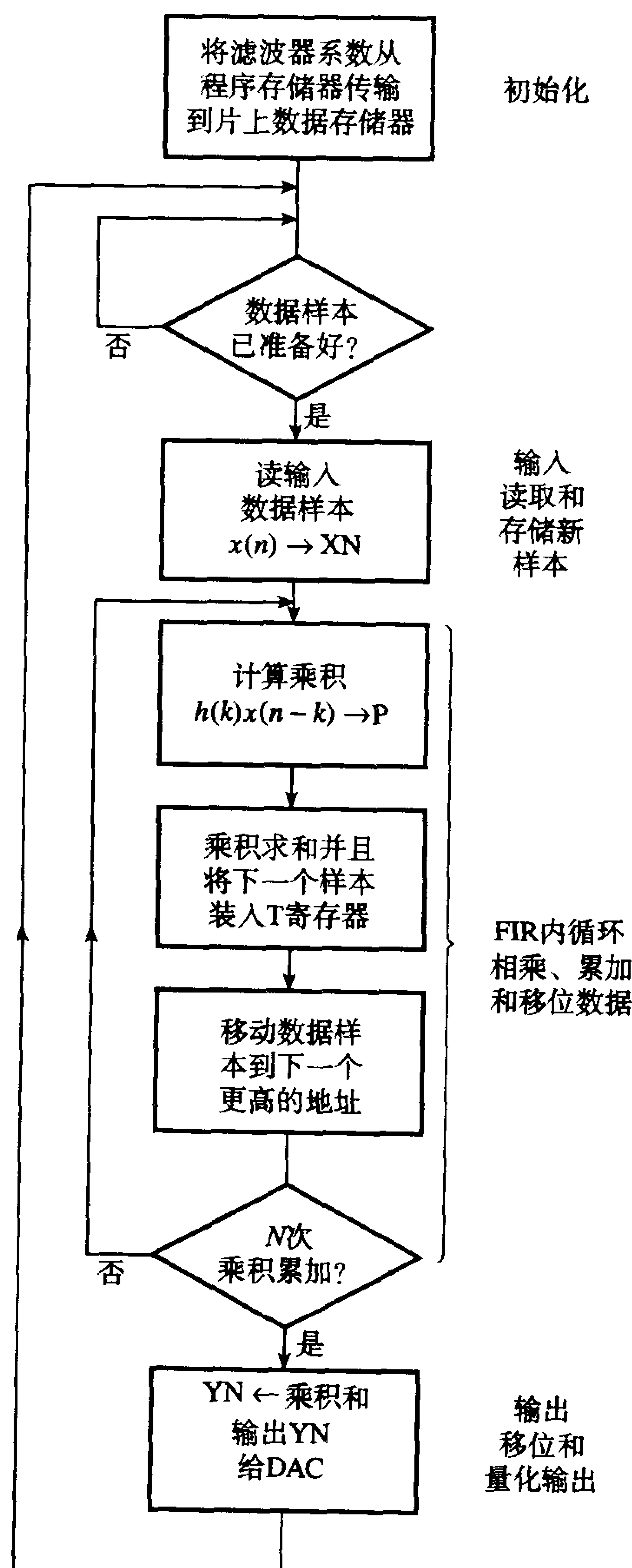


图 12.25　FIR 滤波器的流程图。FIR 内循环执行 12.4 式中的卷积求和

在第二代和以后的DSP处理器中实现 N 点 FIR 滤波器的一个可选择的方法是使用环形缓冲器(circular buffer)。显然在FIR 滤波中，系数存储器的内容是静态的，但是当每个新输入数据样本到达的时候数据存储器是变化的。连续的新数据样本送至一个滑动窗同时最早的数据样本落下来，这是非常有效的。环形缓冲器可以用于处理用于FIR 滤波的输入数据样本块中的变化，而且不需要像在线性数据缓冲器中那样移动数据。

从概念上讲，环形缓冲器和线性缓冲器是一样的，如果我们考虑两端相邻的线性缓冲器，比如最新的和最早的数据样本 $x(n)$和 $x[n-(N-1)]$是相邻的，请参见图 12.26(a)。在图 12.26 中的环形缓冲器中，数据指针（用箭头表示）指向最新的输入样本位置$x(n)$，之前的输入数据样本$x(n-1)$, $x(n-2)$,..., $x(n-7)$存储在连续的位置，按顺时针方向。FIR内循环在每个抽样周期执行，和以前一样，将每个数据样本和对应的滤波器$h(k)$系数相乘，然后累加乘积。惟一的不同是不移动数据样本。内循环计算之后，指针位于$x(n-7)$，即最早的数据样本，然后被下一个输入样本$x(n)$覆盖。图 12.26(a)～图 12.26(c)解释了对于三个连续的数据样本环形缓冲器是如何工作的。

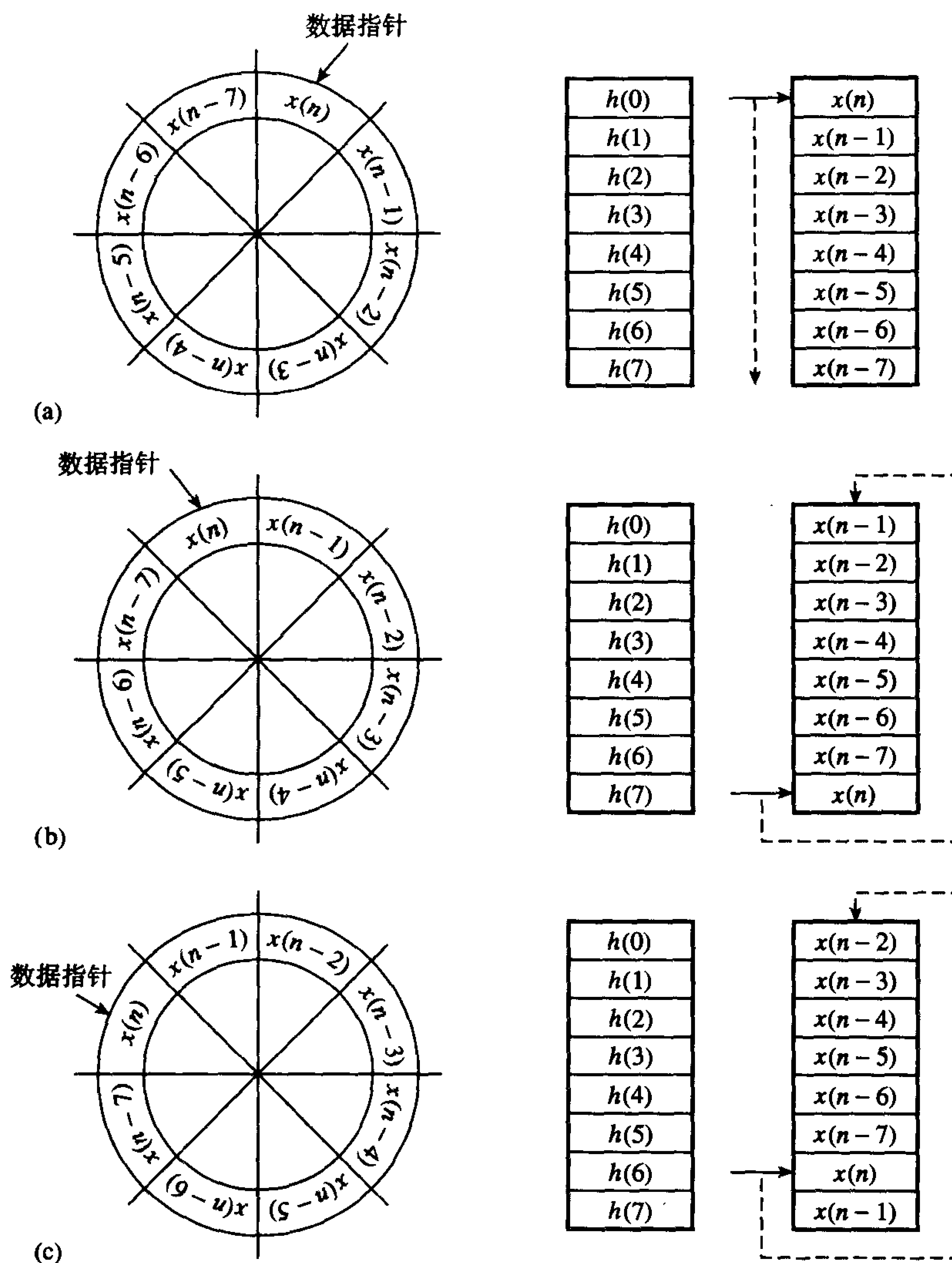


图 12.26　基于环形缓冲器的 FIR 实现原理的解释

实际上，环形寻址是通过使用模（modulo）算术，当地址指针落在缓冲器范围之外时产生一个自动的回绕（wraparound）来实现的。通常，我们需要规定环形缓冲器的起始地址和缓冲器大小（或模大小）。使用环形寻址的 N 点 FIR 滤波器的内循环的一个 DSP56000 实现如下所示。

```
MOVE      #XDATA, R0
MOVE      #COEFF, R4
MOVE      #N-1, M0                    ; buffer/modulo size
```

```
MOVEP   X: INPUT, X: (R0)               ; read and store input sample
CLR     A                               ; clear the accumulator
REP     #N-1                            ; execute FIR inner loop
MAC     X0, Y0, A      X:(R0)+, X0      Y:(R4)+, Y0
MACR    X0, X0, A      (R0)-
```

在本例中，环形缓冲器用于存储数据和系数。环形数据缓冲器执行上面隐含的时移。然而，环形系数缓冲器在这里比较方便地用于系数指针的自动回绕。上面头四条指令设置地址指针R0和R4。FIR内循环由指令REP和MAC执行。重复指令REP重复下一条指令$N-1$次。下一条指令利用了DSP56000的多通道体系结构和并行性，执行一组多操作（multiple operations）——将X0和Y0中的数据和系数相乘，将乘积加到累加器，从X和Y存储器提取下一组相乘的数据和系数，更新指针。

除了FIR滤波，环形缓冲器对于很多要求时移或FIFO队列的DSP函数的有效实现都是很有用的，例如相关、多速率滤波器（抽取和差值滤波器）和周期波形产生。它的使用消除了移动数据或地址指针持续检查/重置的需要。新一代的DSP处理器已经增强了环形寻址的能力。

例 12.3　在第二代定点DSP处理器TMS320C50上实现一个满足下面给定指标的数字FIR陷波滤波器。

陷波频率	1.875 kHz
陷波衰减	60 dB
通带边频	1.575 kHz 和 2.175 kHz
通带波纹	0.01 dB
抽样频率	7.5 kHz

61点的最佳FIR滤波器满足以上指标。这个滤波器的设计在7.6.5节有详细讨论。这里，我们将只集中于讨论实现。滤波器的系数量化为16位（Q15格式），这通过将每个系数乘以2^{15}，然后近似为最接近的整数实现。量化和非量化的系数列于表12.3。如图12.25中的流图所显示的，完整的FIR滤波器至少有四个基本部分：

(1) **初始化**　初始化系统；这可能包括设置系数表。

(2) **输入部分**　这可能包括比如通过串口从ADC读取输入样本$x(n)$。

(3) **内循环计算**　执行FIR公式计算$y(n)$。

(4) **输出部分**　这可能包括内循环计算结果的移位/舍入，将结果比如通过串口送给DAC。

表 12.3　例 12.3 的滤波器系数

	量化的系数
FILTER LENGTH = 61	
***** IMPULSE RESPONSE *****	
H(1) = 0.12743640E-02 = H(61)	42
H(2) = 0.26730640E-05 = H(60)	0
H(3) = -0.23681110E-02 = H(59)	-78
H(4) = -0.17416350E-05 = H(58)	0
H(5) = 0.43428480E-02 = H(57)	142
H(6) = 0.53579250E-05 = H(56)	0
H(7) = -0.71570240E-02 = H(55)	-235
H(8) = -0.49028620E-05 = H(54)	0
H(9) = 0.10897540E-01 = H(53)	357
H(10) = 0.89629280E-05 = H(52)	0
H(11) = -0.15605960E-01 = H(51)	-511
H(12) = -0.85508990E-05 = H(50)	0
H(13) = 0.21226410E-01 = H(49)	695

（续表）

	量化的系数
H(14) = 0.12250150E–04 = H(48)	0
H(15) = –0.27630130E–01 = H(47)	–905
H(16) = –0.11091200E–04 = H(46)	0
H(17) = 0.34579770E–01 = H(45)	1133
H(18) = 0.13800660E–04 = H(44)	0
H(19) = –0.41774130E–01 = H(43)	–1369
H(20) = –0.11560390E–04 = H(42)	0
H(21) = 0.48832790E–01 = H(41)	1600
H(22) = 0.12787590E–04 = H(40)	0
H(23) = –0.55359840E–01 = H(39)	–1814
H(24) = –0.90065860E–05 = H(38)	0
H(25) = 0.60944450E–01 = H(37)	1997
H(26) = 0.88997300E–05 = H(36)	0
H(27) = –0.65232190E–01 = H(35)	–2137
H(28) = –0.38167120E–05 = H(34)	0
H(29) = 0.67925720E–01 = H(33)	2226
H(30) = 0.27041150E–05 = H(32)	0
H(31) = 0.93115220E+00 = H(31)	30512

因为步骤1、2和4是和系统相关的，我们这里集中于讨论内循环计算。FIR内循环在TMS320C50中可以使用下列指令实现：

```
SACL    XN            ; store newest sample, x(n), in data memory
LAR     AR1, #XNM1    ; point to location of oldest data sample, x[n–(N–1)]
ZAP                   ; clear the accumulator and product register
MAR     *, AR1        ; make AR1 current auxiliary register
RPT     #60           ; execute FIR inner loop
MACD    #COEFF, *–    ; multiply and accumulate with data shift
APAC                  ; add last product
```

在本例中，系数和数据存储器如图12.24(c)显示的那样进行组织。辅助寄存器AR1用于内循环计算中的间接寻址（指令MACD），初始化时指向数据存储器中最早的数据样本XNM1。在内循环中，MACD指令进行了下列工作：

- 将前一个乘积加到累加器——初始化时，乘积是零；
- 将系数 $h(k)$ 和AR1指向的数据相乘——初始化时，$h(k) = h(N-1)$，辅助寄存器指向 $x[n-(N-1)]$；
- 将AR1指向的数据复制到下一个更高的位置——初始化时，$x[n-(N-1)]$ 复制到 $x(n-N)$；也就是最早的数据丢弃。最后的MACD指令将 $x(n)$ 复制到 $x(n-1)$，为下一个输入样本腾出空间；
- 将AR1减1（也就是指向数据存储器中的下一个样本）——初始化时，AR1指向 $x[n-(N-1)]$，然后顺序指向 $x[n-(N-2)], x[n-(N-3)], \ldots, x(n)$，如同我们在循环中行进时；
- 将COEFF地址加1——连续的地址是 $h(N-1), h(N-2), \ldots, h(0)$。

12.5.2 IIR数字滤波

12.5.2.1 IIR滤波的基本构造块

二阶IIR滤波器节形成了数字IIR滤波器的基本构造块。两个使用最广的二阶结构是标准（canonic）节（参见图12.27）和直接形式（参见图12.28）。标准二阶节由下面的公式表征：

$$w(n) = SF_1 x(n) - a_1 w(n-1) - a_2 w(n-2) \quad (12.6a)$$

$$y(n) = b_0 w(n) + b_1 w(n-1) + b_2 w(n-2) \quad (12.6b)$$

这里$x(n)$表示输入数据，$w(n)$表示内部节点，$y(n)$是滤波器的输出样本，SF_1是比例因子、等于$1/s_1$。直接形式二阶 IIR 滤波器（参见图 12.28(b)）节的差分方程由下式给出：

$$y(n) = b_0x(n) + b_1x(n-1) + b_2x(n-2) - a_1y(n-1) - a_2y(n-2) \quad (12.7)$$

这里$x(n-k)$是输入数据序列，$y(n-k)$是输出数据序列。直接结构的数据和系数存储在图 12.28(b)中描述。

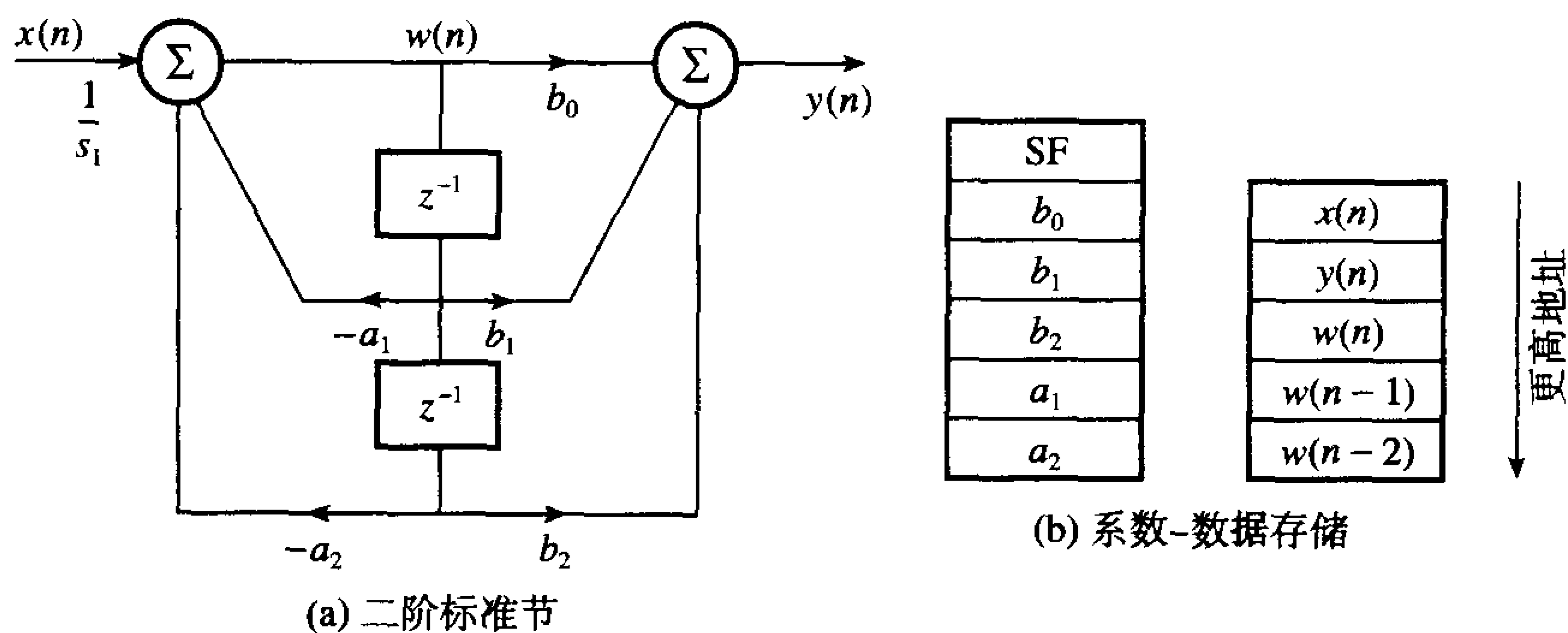


图 12.27　二阶标准节

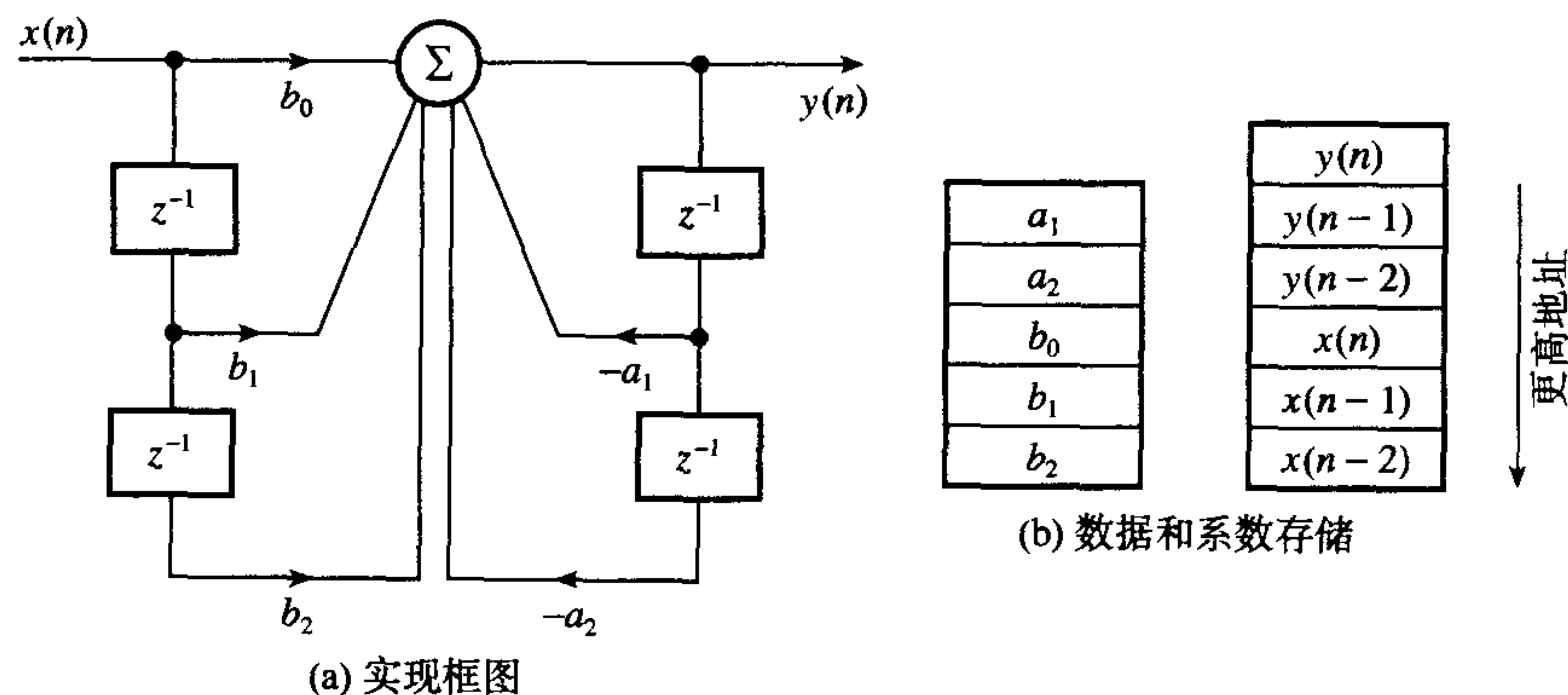


图 12.28　直接形式二阶节的实现

直接形式滤波器比标准节编程更简单，能够更快一点实现，因为涉及的索引更简单：例如，比较 12.6 式和 12.7 式。

高阶 IIR 滤波器可以通过二阶滤波器节的串联（cascade）或并联（parallel）组合实现（更多细节请参见第 8 章）。

串联实现

使用串联的二阶节实现的N阶 IIR 滤波器的传输函数$H(z)$由下式给出：

$$H(z) = \prod_{k=1}^{N/2} \frac{b_{0k} + b_{1k}z^{-1} + b_{2k}z^{-2}}{1 - a_{1k}z^{-1} - a_{2k}z^{-2}} \quad (12.8)$$

使用二阶标准节串联实现的四阶（$N=4$）IIR 滤波器如图 12.29(a)所示。滤波器变量（数据和系数）的存储如图 12.29(b)所示。使用标准节实现的四阶 IIR 滤波器的差分方程组在下面给出：

$$w_1(n) = SF_1x(n) - a_{11}w_1(n-1) - a_{21}w_1(n-2) \quad (12.9a)$$

$$y_1(n) = b_{01}w_1(n) + b_{11}w_1(n-1) + b_{21}w_1(n-2) \quad (12.9b)$$

$$w_2(n) = y_1(n) - a_{12}w_1(n-1) - a_{22}w_2(n-2) \tag{12.9c}$$

$$y_2(n) = b_{02}w_2(n) + b_{12}w_2(n-1) + b_{22}w_2(n-2) \tag{12.9d}$$

二阶标准节串联实现的 IIR 滤波器的 C 语言伪代码在程序 12.3 给出。

由 M 个双二阶（biquadratic）节（这里 $M = N/2$）串联组成的 N 阶 IIR 滤波器的 TMS320C50 和 DSP56000 实现片断分别在程序 12.4 和程序 12.5 中给出。

(a) 实现框图

系数
b_{01}
b_{11}
b_{21}
a_{11}
a_{21}
b_{02}
b_{12}
b_{22}
a_{12}
a_{22}

数据
$x(n)$
$y_1(n)$
$w_1(n)$
$w_1(n-1)$
$w_1(n-2)$
$y_2(n)$
$w_2(n)$
$w_2(n-1)$
$w_2(n-2)$

(b) 数据和系数存储

图 12.29 IIR 滤波器的串联实现

程序 12.3 串联 IIR 滤波器的 C 语言伪代码

```
for(n=0; n<(Nsamples-1); ++n){                /* Nsamples no of data samples */

    xn=x[n];
    for(k=1; k<N; ++k){
        wk=sk[k]*xn-a1[k]*w1[k]-a2[k]*w2[k];
        yk=(b0[k]*wk+b1[k]*w1[k]+b2[k]*w2[k]);
                                              /* output of 1st section */
        w2[k]=w1[k];
                                              /* shift and save delay node data */
        w1[k]=wk;
        xn=yk;                                /* kth section feeds next section */
    }
    y[n]=yk;                                  /* nth output sample */
}
```

程序 12.4　*M*个串联的二阶标准滤波器节的 TMS320C50 实现

```
         SPLK    #M–1, BRCR      ; no. of biquadratic sections
         RPTB    M_IIR           ; compute M biquads in cascade
         LT      *–, AR2         ; load wk(n–2)
         MPYA    +, AR1          ; compute wk(n–2)*ak(2)
         MPY     +               ; wk(n–1)*ak(1)
         LTA     *+, AR1         ; compute and save wk(n)=x(n)+
                                 ; wk(n–1)*ak(1)+wk(n–2)*ak(2)
         SACH    *0+, 1
         MPY     *–              ; compute wk(n–2)*bk(2)
         LACL    #0                ;
         LTD     *–, AR2         ; shift data wk(n–2) = wk(n–1)
         MPY     *+, AR1         ; yk=yk+wk(n–2)*bk(2), wk(n–1)*bk(1)
         LTD     *–, AR2         ; shift data wk(n–1) = wk(n)
         MPY     *+, AR1         ;compute wk(n–2)*bk(2) +
                                 ; wk(n–1)*bk(1), wk(n)*bk(0)
M_IIR    :
         LTA     *, AR4          ; add last product
         SACH    *, 1            ; quantize and save output sample
```

程序 12.5　*M*个串联的二阶标准滤波器节的 DSP56000 实现

```
        DO     #M, M_IIR                                   ; compute M biquads
        MAC    –X0, Y0, A   X: (R0)–, X1   Y: (R4)+, Y0    ;
        MACR   –X1, Y0, A   X1, X: (R0)+   Y: (R4)+, Y0    ; shift data
                                                           ; w(n–2)=w(n–1)
        MAC    X0, Y0, A    A, X: (R0)+    Y: (R4)+, Y0
        MAC    X1, Y0, A    X: (R0)+, X0   Y: (R4)+, Y0
M_IIR
        RND
        MOVEP A, Y: OUTPUT
```

例 12.4

(1) 使用 TMS320C50 定点 DSP 处理器，设计和实现一个低通 IIR 数字滤波器，满足下列指标：

抽样频率	15 kHz
通带	0 ~ 3 kHz
过渡带宽	450 Hz
通带波纹	0.5 dB
阻带衰减	45 dB

(2) 使用 TMS320C54 定点 DSP 处理器重复(1)。

(3) 分别使用 DSP56000 和 DSP56300 重复(1)和(2)。

解：

这个滤波器的详细设计已经在第 8 章（参见 8.8 节）给出。在那里说明了具有下面传输函数的四阶椭圆滤波器能够满足指标：

$$H(z) = \frac{1 + 0.675\,718z^{-1} + z^{-2}}{1 - 0.495\,935z^{-1} + 0.761\,864z^{-2}} \times \frac{1 + 1.649\,656z^{-1} + z^{-2}}{1 - 0.829\,328z^{-1} + 0.307\,046z^{-2}}$$

使用标准节串联实现的差分方程和 12.9 式相同。

为了避免溢出，缩放后的数值、量化为 16 位后的数值以及相应的系数列于表 12.4。由于缺少空间，使用标准形式滤波器组成的四阶滤波器的 TMS320C50 和 TMS320C54 代码没有在这里列出，但是可以在配套的 CD 上和指导手册中找到（细节参见前言）。对于 DSP56000 和 DSP56300 的实现，系数量化为 24 位的处理器字长。这些代码也可以在 CD 上找到。

表 12.4 量化为 16 位之前和之后的滤波器系数

	系数	缩放后	量化后
b_{02}	1	0.999 969 5	32 767
b_{12}	0.675 718	0.675 718	22 142
b_{22}	1	0.999 969 5	32 767
a_{12}	−0.495 935	−0.495 935	−16 251
a_{22}	0.761 864	0.761 864	24 965
b_{01}	1	0.131 113 6	4 296
b_{11}	1.649 656	0.216 292 4	7 087
b_{21}	1	0.131 113 6	4 296
a_{11}	−0.829 328	−0.829 328	−27 175
a_{21}	0.307 046	0.307 046	10 061

$s_1 = 2.479\ 158(L_1)$；SF = 0.403 362 7；$s_2 = 18.908\ 47(L_1)$。

并联实现

并联实现的 N 阶 IIR 滤波器的传输函数由下式给出：

$$H(z) = \prod_{k=1}^{N/2} \frac{b_{0k} + b_{1k}z^{-1}}{1 + a_{1k}z^{-1} + a_{2k}z^{-2}} + C \qquad (12.10)$$

对于$N = 4$，使用二阶标准节的实现框图在图12.30中给出。对于标准节，差分方程在下面给出:

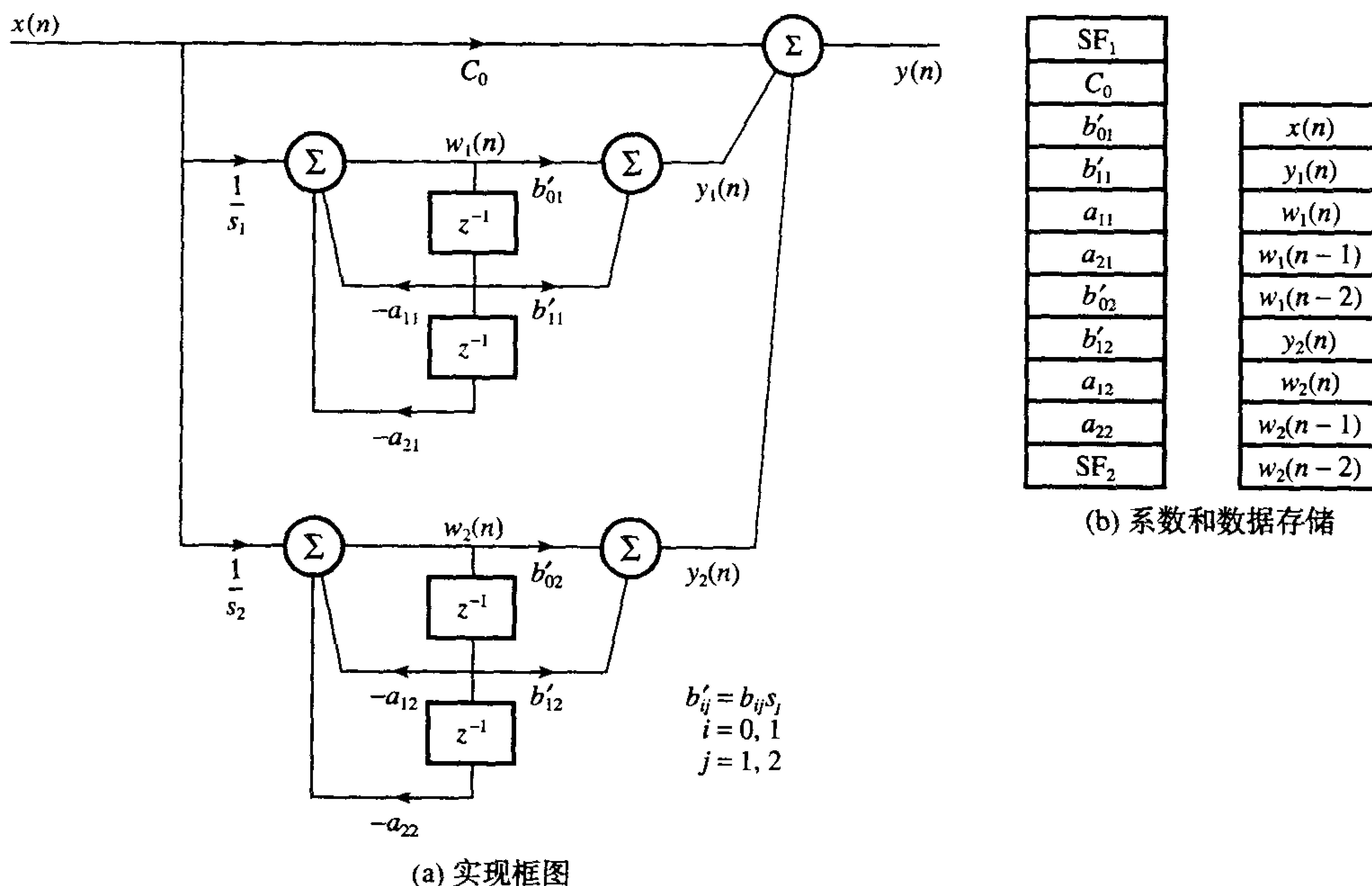
$$w_1(n) = \mathrm{SF}_1 x(n) - a_{11}w_1(n-1) - a_{21}w_1(n-2)$$

$$y_1(n) = b_{01}w_1(n) + b_{11}w_1(n-1)$$

$$w_2(n) = \mathrm{SF}_2 x(n) - a_{12}w_2(n-1) - a_{22}w_1(n-2)$$

$$y_2(n) = b_{02}w_2(n) + b_{12}w_2(n-1)$$

$$y(n) = c_0 x(n) + y_1(n) + y_2(n)$$



(a) 实现框图

(b) 系数和数据存储

图 12.30 一个四阶 IIR 滤波器的实现

由二阶标准节并联组合实现的一个 IIR 滤波器的一个简单 C 语言代码在程序 12.6 中给出。

程序 12.6　并联实现的 C 语言伪代码

```
for(n=0; n<(Nsamples-1); ++n){                /* Nsamples no of data samples */
    y[n] = c*x[n];                            /* output through constant path */
    for(k=1; k<N; ++k){
        wk=sk[k]*x[n]-a1[k]*w1[k]-a2[k]*w2[k];
        yk=(b0[k]*wk+b1[k]*w1[k])/sk[k];      /* output of 1st section */
        w2[k]=w1[k];                          /* shift and save delay node data */
        w1[k]=wk;
        y[n]=yk+y[n];
    }
}
```

例 12.5　使用二阶标准节作为构造块，以并联形式重新表示 12.4 中的传输函数。使用和上一个例子相同的硬件实现滤波器。

解：

使用第 4 章中讨论的部分分式展开（partial fraction expansion）程序，从串联实现中得到并联实现的系数。传输函数变为

$$H(z) = \frac{-0.132\,922\,5 - 0.180\,523\,2z^{-1}}{1 - 0.028\,994z^{-1} + 0.044\,541\,6z^{-2}} + \frac{-0.058\,534 + 0.508\,420z^{-1}}{1 - 0.048\,489\,9z^{-1} + 0.017\,951\,1z^{-2}} + 0.249\,923\,79$$

$$s_1 = 5.524\,484\,4, \quad s_2 = 2.4794$$

表12.5给出了量化为16位之前和之后的系数值。滤波器的TMS320C50、TMS320C54、DSP56000和DSP56300代码可以在配套的CD上和指导手册中找到（细节参见前言）。对于DSP56000和DSP56300，系数量化为24位。

表 12.5　例 12.5 中四阶 IIR 滤波器的实现：量化为 16 位之前和之后的滤波器系数

	未量化系数	量化后系数
SF_1	0.181 00	5 931
C_0	0.249 923 79	8 190
b_{01}	−0.132 922 5	−24 063
b_{11}	−0.180 523 2	−32 670
a_{11}	0.028 994	16 251
a_{21}	−0.044 541 6	−24 965
b_{02}	−0.058 534	−4 756
b_{12}	0.508 420 5	20 653
a_{12}	0.048 489 9	27 178
a_{22}	−0.017 951	−10 061
SF_2	0.403 32	13 216

上面讨论的串联和并联结构的实现技术扩展为高阶IIR滤波器是相当容易的。将二阶构造块作为子例程实现可以得到更紧凑的代码。

12.5.3　FFT 处理

有限数据序列 $x(n)$ 的离散傅里叶变换（DFT）定义为

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}$$

这里 W_N 经常称为旋转因子（twiddle factor），它是一组复系数。

当N很大时，DFT系数$X(k)$的直接计算非常耗时。FFT算法提供了计算$X(k)$的有效方法，能够显著减少计算时间。如在第3章所讨论的，蝶形和旋转因子是FFT算法的中心。

12.5.3.1 蝶形的实现

图12.31(a)和图12.31(b)描述了在基-2 FFT中使用的两种类型的蝶形。基于这些蝶形的FFT产生相同的结果。对于时域抽取（图12.31(a)），采用一对输入数据A和B，产生了一对输出：

$$A' = A + BW_N^k \tag{12.11a}$$

$$A' = A + BW_N^k \tag{12.11b}$$

总体上，输入和输出数据样本以及旋转因子都是复数，能够表示为

$$A = A_r + jA_i \tag{12.12a}$$

$$B = B_r + jB_i \tag{12.12b}$$

$$W_N^k = e^{-j2\pi k/N} = \cos(2\pi k/N) - j\sin(2\pi k/N) \tag{12.12c}$$

这里下标r表示数据的实部，下标i表示数据的虚部。12.11式中的蝶形操作涉及到复数算术，但实际上经常使用实数算术计算。为了将操作表示为适合于实数算术的形式，我们注意到12.11式中B和W的乘积具有这样的形式：

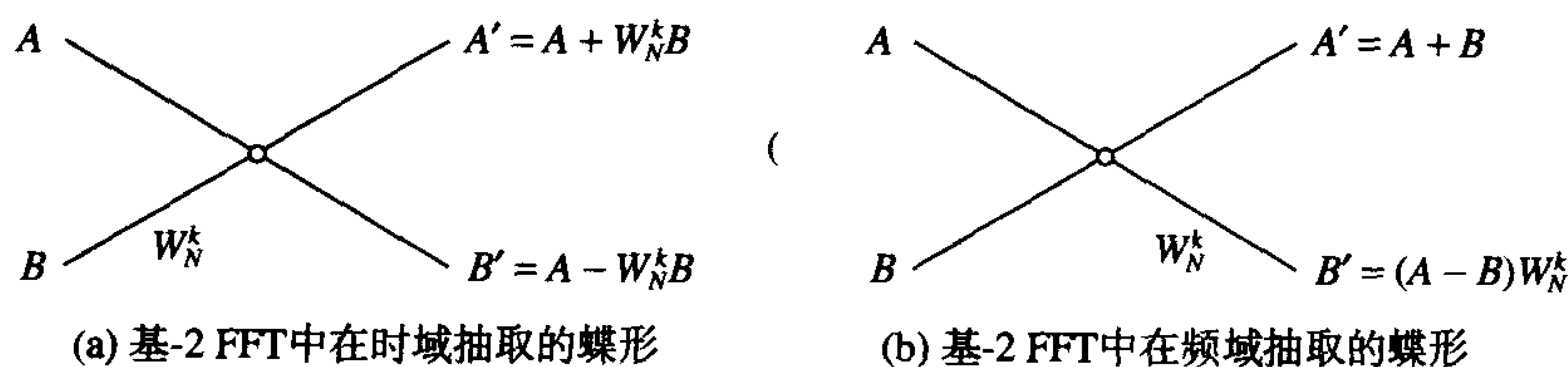
$$BW_N^k = B_r\cos(X) + B_i\sin(X) + j[B_i\cos(X) - B_r\sin(X)] \tag{12.13}$$

这里$X = 2\pi k/N$。在12.11a式和12.11b式中使用12.12式和12.13式，我们有

$$A' = A_r + [B_r\cos(X) + B_i\sin(X)] + j\{A_i + [B_i\cos(X) - B_r\sin(X)]\} \tag{12.14a}$$

$$B' = A_r - [B_r\cos(X) + B_i\sin(X)] + j\{A_i - [B_i\cos(X) - B_r\sin(X)]\} \tag{12.14b}$$

蝶形的输出A'和B'，现在是期望的形式。因此，以矩形形式给出一对复数数据点A和B，12.14a式和12.14b式就可以用于使用实数算术计算蝶形的输出。



(a) 基-2 FFT中在时域抽取的蝶形　　(b) 基-2 FFT中在频域抽取的蝶形

图12.31　在基-2 FFT算法中用到的两种类型的蝶形

12.14式中的正弦和余弦项的计算是很耗时的。在实时FFT中，一种更有效的方法是预先计算出旋转因子（参见12.12c式）的实部和虚部，并将这些数值存储在一个查找表（look-up table）中。程序12.7中的C语言伪代码解释了旋转因子的值是如何预先计算的。

旋转因子预先计算并且存储在查找表中的基-2蝶形的C语言伪代码在程序12.8中给出。

程序12.9中显示的是TMS320C25伪代码。预先计算的旋转因子值以Q15格式存储。输入数据假定为复数，实部和虚部存储在数据RAM中的连续位置。对于复数输入，A'或B'能得到的最大值是2.414 42，对于实数输入是2。在定点算术中，这将产生溢出。为了避免溢出，蝶形的输入数据应该缩放。在C50的实现中，缩放是动态的，利用了两个定点数的乘积产生一个附加符号位的优点。附加符号位一般通过左移消除，但是如果保留它，结果就可以有效地被2缩放。

程序 12.7　预先计算旋转因子值的 C 语言伪代码

```
pi=6.28315307179586/N;
for(k=0; k<N/2; ++k){
	X=k*pi;
	w.real[k]=cos[X];
	w.imag[k]=sin[X];
}
```

程序 12.8　蝶形的 C 语言伪代码

```
t.real=br*w.real[k]+bi*w.imag[k];
t.imag=bi*w.real[k]-br*w.imag[k];
b.real[j]=a.real-t.real;
b.imag[j]=a.imag-t.imag;
a.real[j]=a.real+t.real;
a.imag[j]=a.imag+t.imag;
```

程序 12.9　蝶形的 TMS320C50 代码

```
*
*	compute terms common to the two butterfly outputs, A' and B'
*
*
		LT	BR	;compute 1/2*[b.real*cos(X)+b.imag*sin(X)]
		MPY	WREAL	;1/2*b.real*cos(X)
		LTP	BI
		MPY	WIMAG	;1/2*b.imag*sin(X)
		APAC		;1/2[b.real*cos(X)+b.imag*sin(X)]
		MPY	WREAL	;1/2*b.imag*cos(X)
		LT	BR
		SACH	BR	;1/2[b.real*cos(X)+b.imag*sin(X)]
		PAC		;compute [q.imag*cos(X)-q.real*sin(X)]
		MPY	WIMAG
		SPAC
		SACH	BI
*
*	compute and save the butterfly outputs
*
		LACC	AR, 14	;compute and save the real parts of the output
		ADD	BR, 15
		SACH	AR, 1	;save a.real
		SUB	BR
		SACH	BR, 1	;save b.real
		LAC	AI,14	;compute and save the imaginary parts of the
				;output
		ADD	BI, 15
		SACH	AI, 1	;save a.imag
		SUB	BI
		SACH	BI, 1	;save b.imag
```

12.5.3.2　原位计算和不变几何

图 12.32 显示了一个八点 FFT 的信号流图。从图中可以看出，为了得到显示在右手边的 DFT 系数 $X(k)$，则要给出输入数据，必须执行一系列的蝶形计算。基-2 FFT 算法是一种以一种有序方式执行这一系列蝶形计算的方法。在流图中，数据从左边流向右边。因此，一旦计算出蝶形的输出 A' 和 B'，输入 A 和 B 就不再需要，所以可以被输出覆盖。这就是原位（in-place）计算的基本概念。

原位算法使可用存储器的使用非常有效，因为变换后的数据覆盖了输入数据。在过去，存储器是非常昂贵的，这是一个非常重要的考虑。然而，原位计算所要求的、用来确定在存储器中什么地方提取输入数据给每个蝶形的下标是非常复杂的。例如，图12.32中阶段1顶部的蝶形从地址0和4获取它的输入，并且将结果写回相同的地址。另一方面，阶段2顶部的蝶形从地址0和2获取它的输入。总之，在原位FFT中，输入/输出地址随阶段的不同而改变。进一步，对于高速FFT，对输入和输出使用相同的存储器降低了计算速度，这是因为长时间的存储器访问（使用双端口RAM的例子除外）。因为存储器和乘法器都是廉价的，现在的趋势是优化整个FFT处理器从而提高速度。

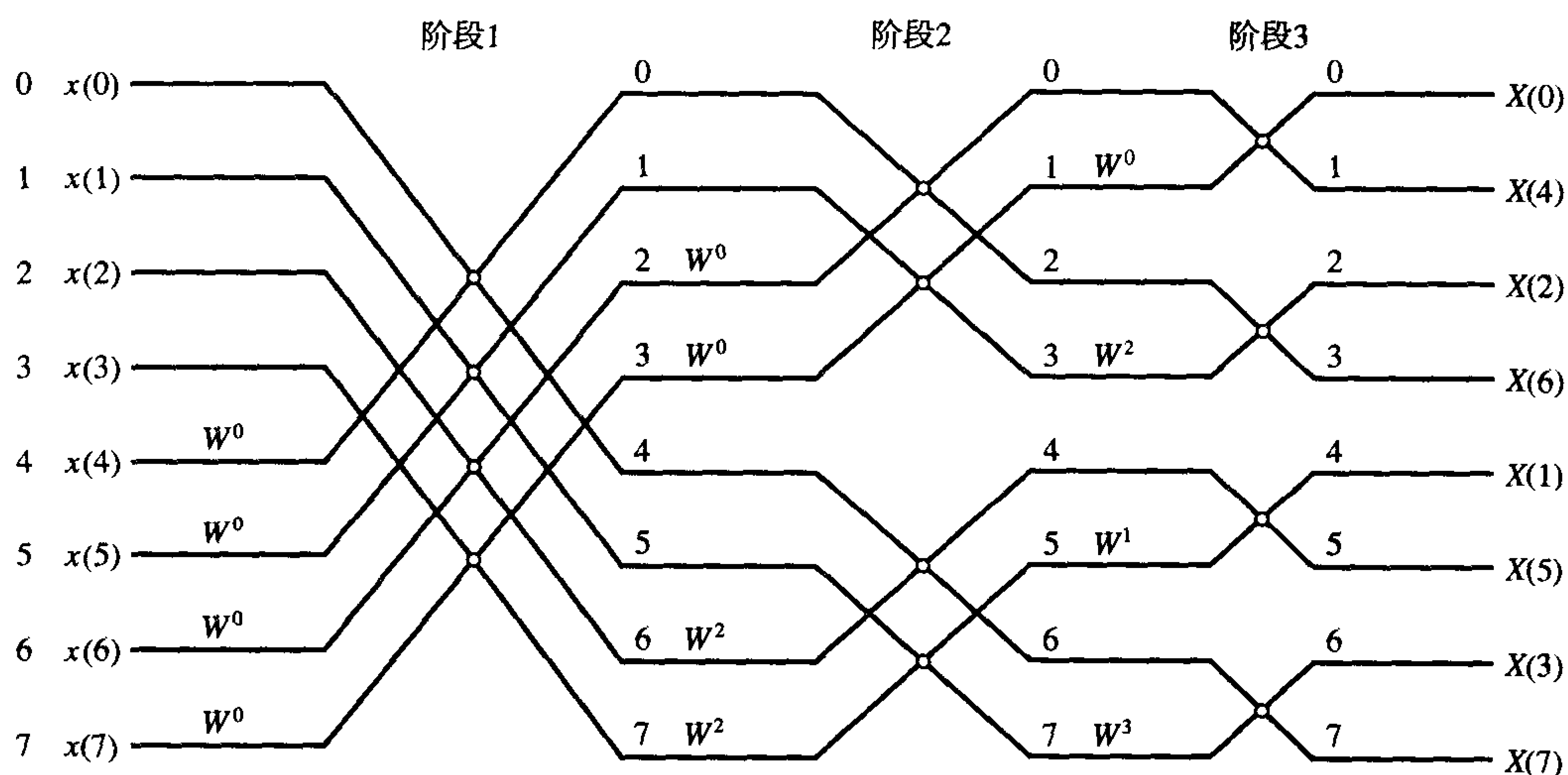


图12.32 一种原位DIT FFT流图，输入是自然顺序但是输出是倒位顺序

一种可选择的FFT实现方法，称为非原位（non-in-place）或不变几何（constant geometry），从一对地址读取输入数据给蝶形，将输出存储在另一对地址，如图12.33所示。和原位FFT不同，其每个蝶形的输入/输出地址随阶段而改变，在此情况下每个蝶形的地址是固定的且简单得多。对于N点FFT，在每个阶段第n个蝶形的输入是$2n$和$2n+1$，其中$n=0,1,\ldots,N/2-1$。第n个蝶形的输出存储在地址n和$N/2+n$。例如，在图12.33的第二阶段，顶部的蝶形从地址0和1获取它的输入，将它的输出存储在地址0和4。很清楚，非原位FFT的工作要求两个分离的存储器或阵列：一个保存输入，另一个保存输出。在每个阶段后，存储器的角色互换。

12.5.3.3 数据搅乱和倒位

在DIT（时域抽取）FFT中，如果输入数据序列以自然顺序应用于FFT处理器，FFT的输出似乎是搅乱的（参见图12.32）。为了保证输出以正确的顺序（即以$X(0), X(1),\ldots, X(N-1)$的顺序）出现，我们或者在进行FFT之前搅乱输入数据序列（参见图12.33和图12.34），或者在进行FFT之后归整（unscramble）输出（参见图12.32）。

对于基-2 FFT，输入数据搅乱是通过以倒位顺序存储输入序列实现的。假定输入数据已经按照自然顺序存储（即以$x(0), x(1), x(2),\ldots, x(N-1)$的顺序），倒位顺序是通过将输入数据下标表示为二进制，对于八点FFT如表12.6第二列所示，然后交换中心左右的位（表12.6中的第四列）。例如，注意到表中数据样本$x(3)$的下标具有二进制表示001。交换中间位左右的第一位和第三位给出110的表示（即位1的0变为1，位3的1变为0，而我们在其左右执行操作的中间位保持不变）。这个倒位代码110是十进制6。为了达到搅乱的效果，我们交换数据样本$x(3)$和$x(6)$的位置。将同样的原理应用于剩余的输入数据，我们得到表12.6第三列给出的倒位序列。注意到在搅乱之后，第一和最后

一个数据样本的位置保持不变。这是因为下标000和111的倒位没有任何效果。总之，基-2 FFT中第一和最后一个数据点不受搅乱的影响。细心的读者将会观察到输入序列中对倒位免疫的其他样本。

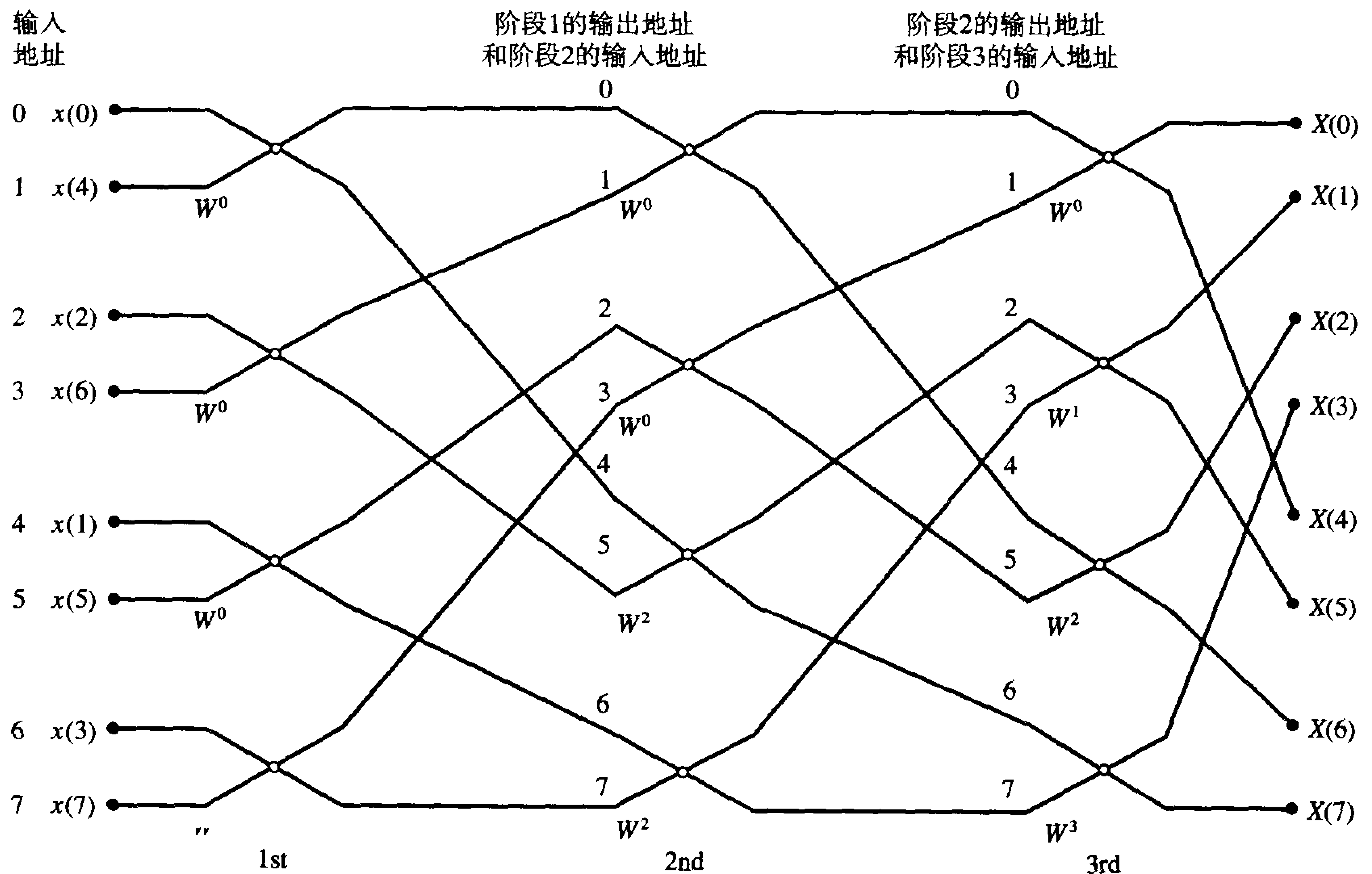


图 12.33　不变几何基-2 FFT。在不变几何中，计算不是原位的。注意到对于每个蝶形输入和输出，间隔是不变的

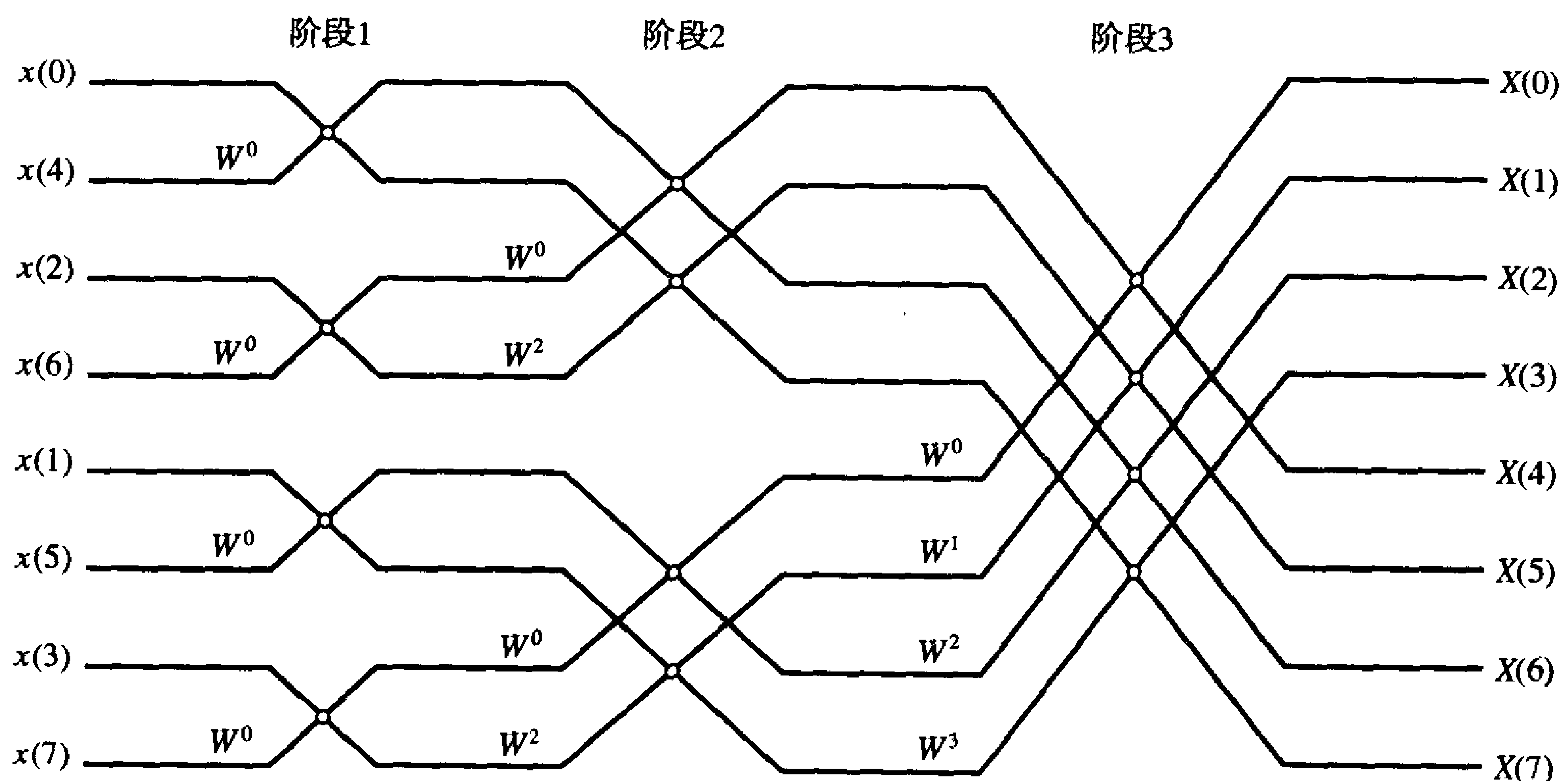


图 12.34　一个原位DIT FFT流图，输入是倒位顺序，输出是自然顺序

当输入数据保存在存储器或阵列中时，搅乱输入数据涉及到识别成对的输入数据位置和互换，或者交换这些位置上的数据。Rader的倒位算法（Rabiner and Gold，1975）被广泛地用来确定要交换的存储器地址的下标。倒位算法的一个C语言伪代码在程序12.10中给出。

表 12.6 显示倒位概念的八点 FFT 的数据

输入序列，自然顺序	序列的二进制代码	输入序列，倒位	序列（倒位）的二进制代码
$x(0)$	000	$x(0)$	000
$x(1)$	001	$x(4)$	100
$x(2)$	010	$x(2)$	010
$x(3)$	011	$x(6)$	110
$x(4)$	100	$x(1)$	001
$x(5)$	101	$x(5)$	101
$x(6)$	110	$x(3)$	011
$x(7)$	111	$x(7)$	111

程序 12.10

```
/* perform in-place bit reversal */

j=1;
for(i=1; i<N; ++i){
        if(i<j){
                tr=x.real[j];      /* swap x[j] and x[i] */
                ti=x.imag[j];
                x.real[j]=x.real[i];
                x.imag[j]=x.imag[i];
                x.real[i]=tr;
                x.imag[i]=ti;
                k=N/2;
                while(k<j){
                        j=j-k;
                        k=k/2;
                }
        }
        else {
                k=N/2;
                while(k<j){
                        j=j-k;
                        k=k/2;
                }
        }
j=j+k;
}
```

现在的高级 DSP 芯片提供指令用来对输入数据样本执行倒位，当它们从存储器中提取出来准备进行 FFT 时；或者对变换后的数据执行倒位，当它们在 FFT 之后存入存储器时。

12.5.4 多速率处理

如同在第 9 章中讨论的，多速率处理涉及到以一种以上的抽样速率执行 DSP 操作。多速率处理中的两种基本操作是抽取（抽样率减少）和内插（抽样率增加）。我们将通过一个例子来解释一个实时抽取器的实现。

例 12.6 一个信号的抽样率通过三级抽取处理从 30 kHz 减小到 1 kHz。假定抽取后感兴趣的最高频率是 400 Hz，带内起伏是 0.08 dB，阻带衰减是 50 dB。抽取器在 TMS320C50 上实现。

解：

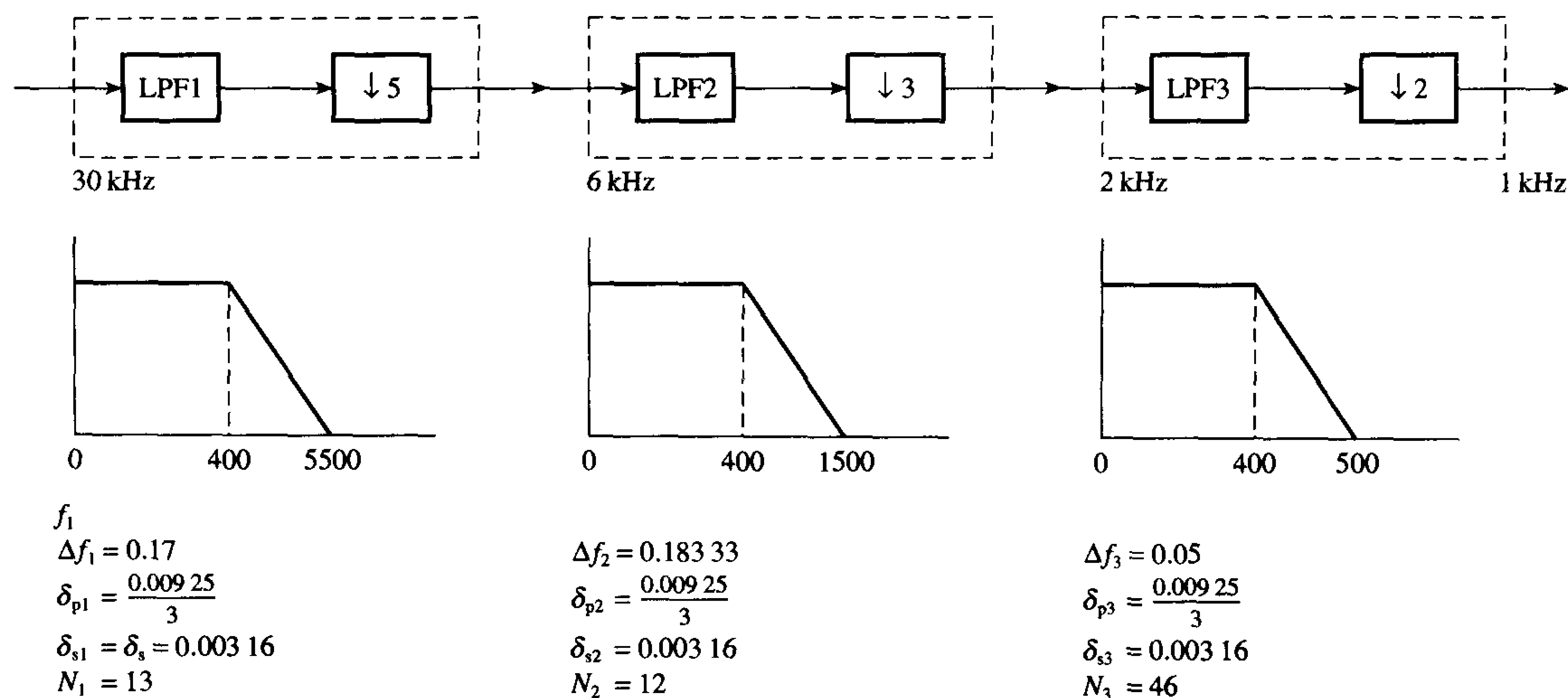
使用多速率设计程序（在配套的 CD 上和指导手册中可以找到，细节见前言），可以得到三级抽取器的参数，参见图 12.35。使用最佳 FIR 设计程序得到的三个滤波器的系数在表 12.7 中给

出。滤波器长度比用抽取器设计程序预测估计的稍长一点（12、13、48 代替了 13、12、46），以保证满足指标。

表 12.7　三级抽取器的滤波器系数

FILTER LENGTH = 12	
***** IMPULSE RESPONSE *****	
H(1) = 0.73075550E–02 = H(12)	239
H(2) = 0.27123260E–01 = H(11)	889
H(3) = 0.59286430E–01 = H(10)	1943
H(4) = 0.10198970E+00 = H(9)	3342
H(5) = 0.14187870E+00 = H(8)	4649
H(6) = 0.16675770E+00 = H(7)	5464
***** IMPULSE RESPONSE *****	
H(1) = –0.86768190E–02 = H(13)	–284
H(2) = –0.25476870E–01 = H(12)	–835
H(3) = –0.25468170E–01 = H(11)	–834
H(4) = 0.24184320E–01 = H(10)	792
H(5) = 0.13238570E+00 = H(9)	4338
H(6) = 0.24907950E+00 = H(8)	8162
H(7) = 0.30075170E+00 = H(7)	9855
FILTER LENGTH = 48	
***** IMPULSE RESPONSE *****	
H(1) = 0.17780220E–02 = H(48)	585
H(2) = –0.17396640E–02 = H(47)	–57
H(3) = –0.49461790E–02 = H(46)	–162
H(4) = –0.25451430E–02 = H(45)	–83
H(5) = 0.40843330E–02 = H(44)	134
H(6) = 0.42773070E–02 = H(43)	140
H(7) = –0.45042640E–02 = H(42)	–148
H(8) = –0.80385180E–02 = H(41)	–263
H(9) = 0.29002500E–02 = H(40)	95
H(10) = 0.12193670E–01 = H(39)	400
H(11) = 0.92281120E–03 = H(38)	30
H(12) = –0.16199860E–01 = H(37)	–531
H(13) = –0.76966970E–02 = H(36)	–252
H(14) = 0.18898710E–01 = H(35)	619
H(15) = 0.17966280E–01 = H(34)	589
H(16) = –0.18756490E–01 = H(33)	–615
H(17) = –0.32451860E–01 = H(32)	–1063
H(18) = 0.13458800E–01 = H(31)	441
H(19) = 0.52945520E–01 = H(30)	1735
H(20) = 0.17620600E–02 = H(29)	58
H(21) = –0.86433440E–01 = H(28)	–2832
H(22) = –0.44585360E–01 = H(27)	–1461
H(23) = 0.18176500E+00 = H(26)	5956
H(24) = 0.41039480E+00 = H(25)	13448

一个通用的三级抽取器的流程图在图 9.15 中给出。基于 TMS320C50 的抽取器的系数和数据存储映像如图 12.36 所示。滤波器系数通过将每个系数乘以 2^{15} 然后近似为最接近的整数而量化为 16 位。使用 TMS320C50 汇编语言编写的抽取器程序是通用的，通过替换系数，规定滤波器长度、抽取级数和抽取因子，可以用于实现一级、两级或三级抽取。



实际的滤波器参数

滤波器1：频带边沿, 0, 0.013 33, 0.183 33
　　　　滤波器长度, 12; 权重, 1, 3

滤波器2：频带边沿, 0, 0.066 66, 0.25, 0.5
　　　　滤波器长度, 13; 权重, 1, 3

滤波器3：频带边沿, 0, 0.2, 0.25, 0.5
　　　　滤波器长度, 48; 权重, 1, 3

图 12.35　三级抽取器的参数

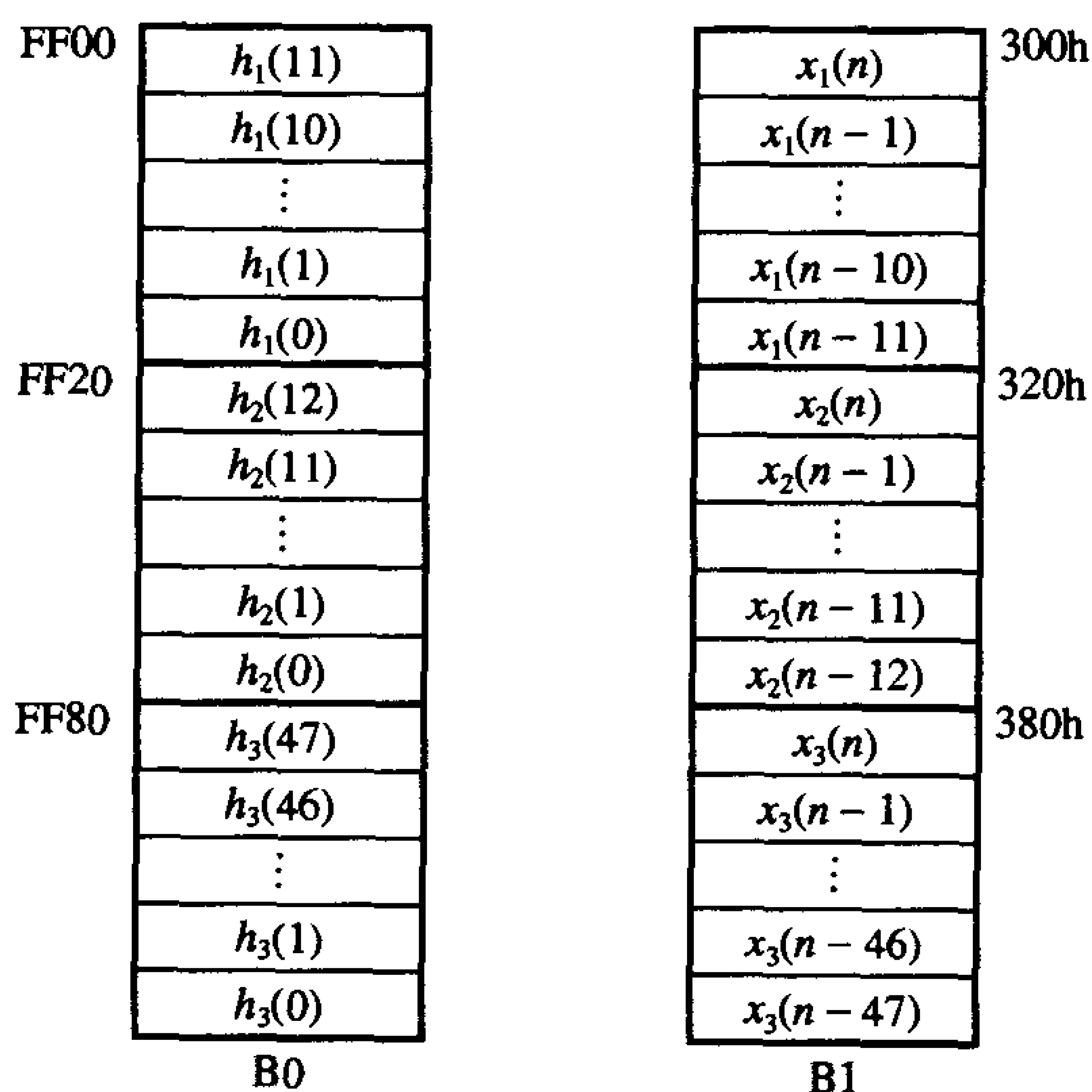


图 12.36　三级抽取器的系数和数据存储映像

12.5.5　自适应滤波

一个自适应滤波器的通用结构在图12.37中描述。如在第10章中所讨论的，自适应滤波包含两个处理过程。

(1) **数字滤波**　图 12.37 中的数字滤波器的系数用来从输入信号 $x(n)$中提取适当信息以产生 $y(n)$。假定采用横向 FIR 结构，滤波器由下式给出：

$$y(n) = \sum_{k=0}^{N-1} w_n(k)x(n-k) \tag{12.15}$$

这里 $w_n(k)$（$k=0, 1,..., N-1$）是数字滤波器系数（经常称为加权），$x(n-k)$（$k=0, 1,..., N-1$）是输入数据序列。

12.15式中给出的数字滤波器的实现和前面讨论的标准FIR滤波器非常相似。因此滤波器的C语言实现将具有相似的形式：

```
y[n]=0;
for(k=0; k<N; k++){
      y[n]=y[n]+wn[k]*xn[k];
}
```

(2) **自适应处理**　这个处理涉及更新，也就是朝着最佳值的方向调整滤波器系数。当使用基本LMS算法时，系数更新如下：

$$w_{n+1}(k) = w_n(k) + 2\mu e(n)x(n-k), \quad k = 0, 1, 2, \ldots, N-1 \tag{12.16}$$

这里 $w_n(k)$是在第 n 个抽样时刻数字滤波器的第 k 个系数，μ 是稳定因子（stability factor），$x(n-k)$是第 k 个延迟线的第 k 个输入数据样本。基本LMS更新方程的C语言实现在程序 12.11 中给出。$2\mu e$[n]项是标量，对于所有系数都相同，所以它只需要计算一次，并且置于循环之外。一个 TMS320C50 实现的自适应处理在程序 12.12 中给出。

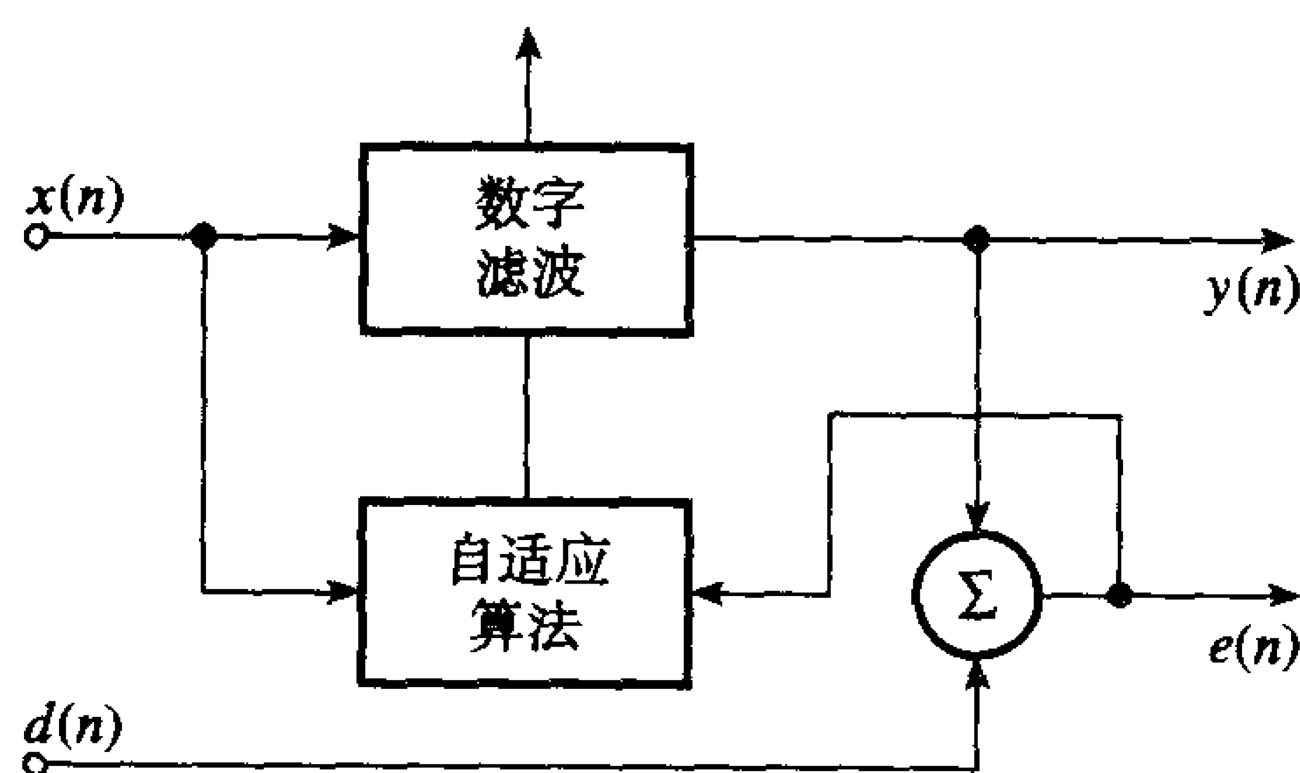


图 12.37　一个自适应滤波器的通用结构：一对输入和一对输出

程序 12.11　LMS 自适应滤波器系数更新的 C 伪代码

```
uen=2*u*e[n]
for(k=0; k<N; k++){
      wn[k]=wn[k]+uen*xn[k];
}
```

程序 12.12　LMS 自适应滤波器系数更新的 TMS320C50 伪代码

```
        LT      ERR
        MPY     U           ;compute μ * e(n)
        PAC
        ADD     ONE, 15
        SACH    ERRF
        LACC    #N-1        ; specify filter length
        SAMM    BRCR
        LAR     AR2, #WNM1  ; point to the last coefficient, wk(N-1)
        LAR     AR3, #XNM1  ; point to x(n-(N-1))
        LT      ERRF
        MPY     -, AR2      ; compute μ e(n) + x(n-k)
        RPTB    LMS-1       ; update coefficients, wk+1(n)
        ZALR    *, AR3
        MPYA    *-, AR2
        SACH    *+
LMS     ZALR    *, AR3
        APAC
        SACH    *+
```

12.6 专用DSP硬件

专用的原因

数字信号处理操作是计算密集型的。在宽带应用中，输入/输出数据率很高，大多数通用数字信号处理器（DSP）不能足够快地执行所要求的计算。这就是通用DSP经常在音频应用的想当然的原因。进一步，对于一个给定的应用，大多数通用DSP包含的很多片上资源是冗余的或是未用的，例如寻址模式，指令集和I/O外围设备。在专用DSP中，硬件是优化过的以执行专门的算法或在专门应用中执行确定的函数。这就使片上资源的使用更充分，操作的速度增加。

专用硬件是作为单片产品或分立元件的模块实现的。使用分立元件的构造模块方法更灵活，能够增加速度，但是硬件开发困难，而且可能更昂贵。单片DSP，如果有适用于该任务的产品，具有较少的芯片数，不需要晦涩的汇编语言知识，也没有软件调试问题。

专用硬件的基本需求

DSP中最常见的算术操作，比如数字滤波、相关和变换是乘积和的形式：

$$y=\sum_{k=0}^{N-1} a_k x_k \tag{12.17}$$

这里a_k是一组系数或变量，x_k是一个数据序列。

特征12.17式可以写成递归的形式，以允许乘积和更有效地计算：

$$y_k=a_k x_k+y_{k-1}, \quad k=0, 1, \ldots, N-1 \tag{12.18}$$

这里

$$y_{-1}=0$$

$$y=y_{N-1}$$

在专用DSP中，12.18式使用乘法-累加器以非常快的速度计算，例如40 ns每MAC。

和通用DSP一样，专用DSP体系结构包括数据存储器，RAM和/或ROM，用于存储数据和变量（比如滤波器或FFT系数）、快速硬件乘法-累加器以及临时寄存器以存储数据或中间结果。充分使用并行技术、多路复用和流水线技术以获得最大的速度。

在下面几节中，我们将讨论DSP专用硬件设计中所涉及的几个基本问题。

12.6.1 硬件数字滤波器

12.6.1.1 FIR数字滤波器

直接形式的FIR滤波器由下面方程表征：

$$y(n)=\sum_{k=0}^{N-1} h(k)x(n-k)$$

图12.38显示了一个使用分立元件模块构造的FIR数字滤波器的基本体系结构。主要元件是系数和数据存储器，模拟输入/输出单元（ADC和DAC）、乘法-累加器（MAC）和一个控制器（没有显示出来）。FIR滤波器的元件可以使用快速的现货产品实现。

在每个抽样时刻，从ADC读取一个新的数据样本$x(n)$，存入数据存储器。每个输入数据样本和对应的系数同时从存储器中提取出来应用于乘法器。然后累加乘积产生输出样本。每个输出样本$y(n)$的计算要求N次数据-系数的存储器提取和N次MAC。

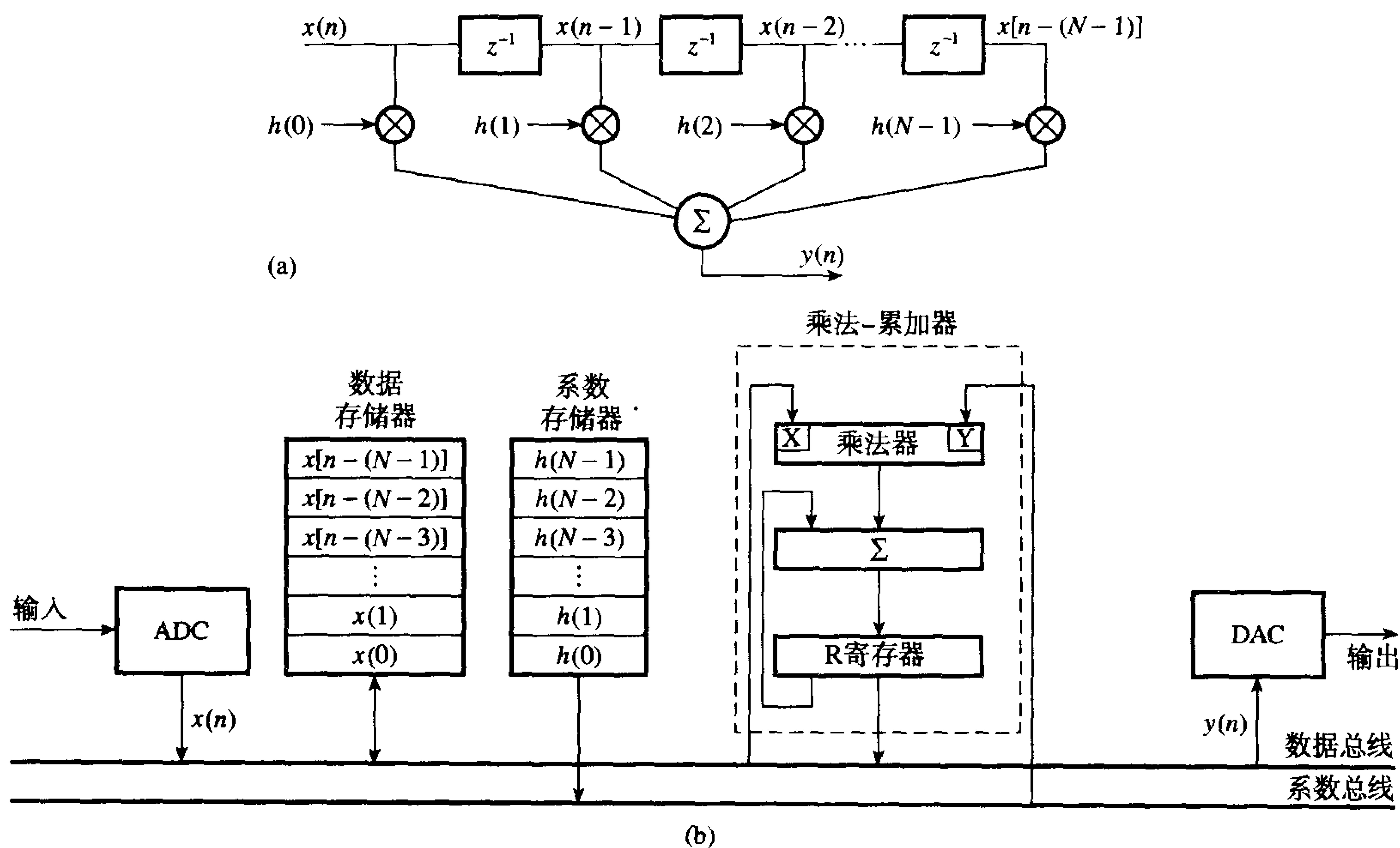
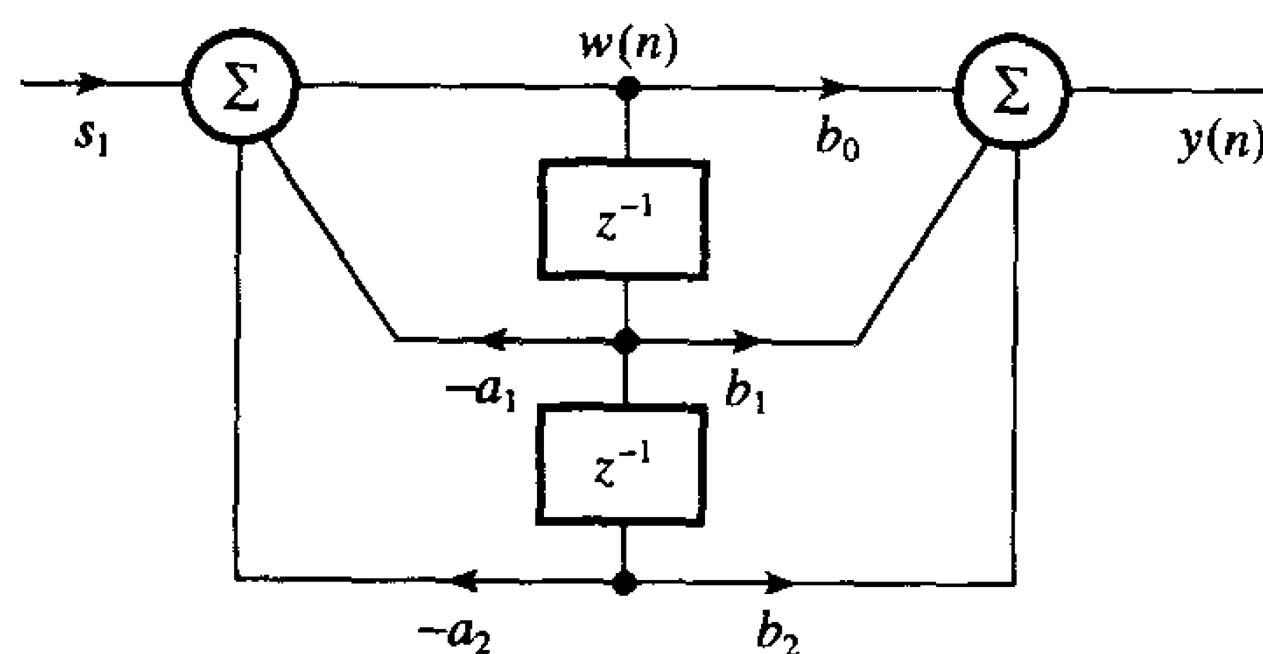


图 12.38　一个硬件 FIR 数字滤波器的体系结构

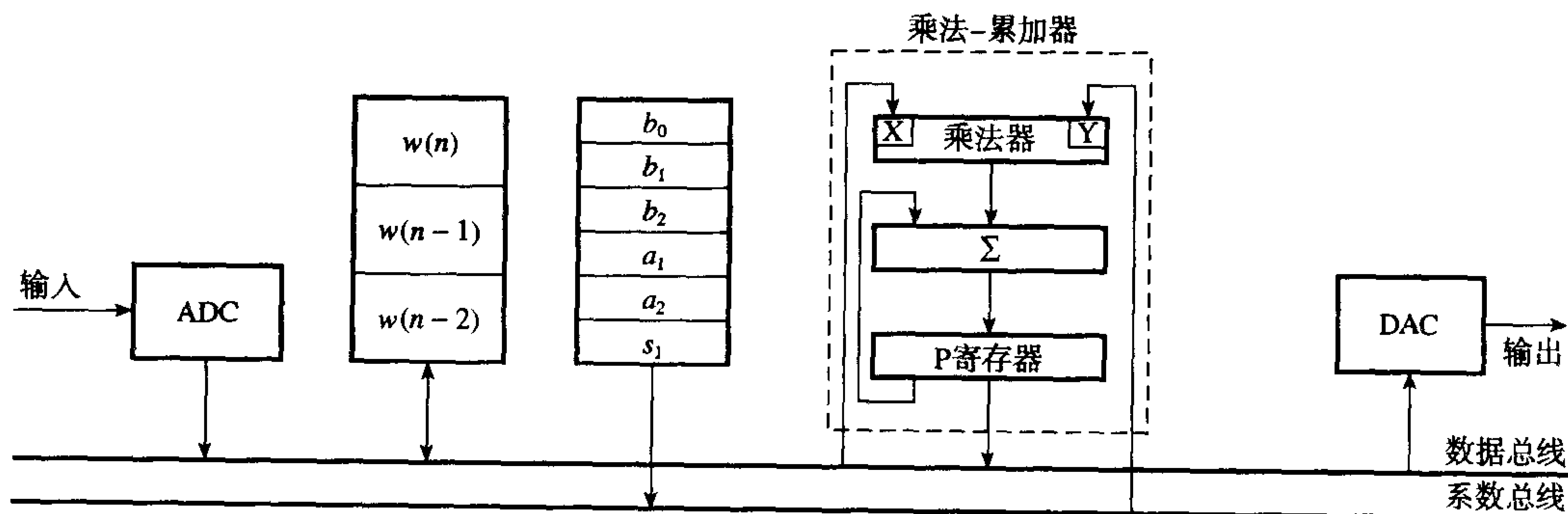
FIR 滤波的操作很规则而且结构很好，可以很容易使用单片 IC 实现。专用单片 FIR 滤波器现在可以买到，比如 Mitel 的 PDSP16256 预先可编程 FIR 滤波器。

12.6.1.2　IIR 数字滤波器

二阶标准节 IIR 滤波器的体系结构如图 12.39 所示。在此例中，数据存储器保存内部节点数据 $w(n)$。图 12.39(a)中的二阶标准节由下列方程表征：



(a) IIR滤波器结构



(b) 一个IIR滤波器双二阶节的硬件体系结构

图 12.39　IIR 滤波器的结构

$$w(n) = s_1x(n) - a_1w(n-1) - a_2w(n-2)$$
$$y(n) = b_0w(n) - b_1w(n-1) - b_2w(n-2)$$

这里 $x(n)$表示输入数据，$w(n)$表示内部节点，$y(n)$是滤波器输出样本，s_1 是比例因子。

12.6.2 硬件 FFT 处理器

DFT 接收一组 N 个时域样本，变换成一组 N 个频域样本 $X(k)$。FFT 是计算 DFT 系数 $X(k)$的一种有效方法。蝶形算术是 FFT 中的基本操作。蝶形由下列方程表征：

$$A' = A + W_N^k B$$
$$B' = A - W_N^k B$$

这里 A 和 B 是一对复数值的蝶形输入数据样本，A' 和 B' 是蝶形的输出，W_N 是旋转因子，也是复数值。

每个蝶形操作要求一个复数乘法（即 W_N^kB）、一个复数加法和一个复数减法。图 12.40 显示了一个蝶形处理器的一种直接硬件实现，使用了复数算术单元的分立模块：一个复数乘法器和一对复数累加器。复数乘法器计算公共项 W_N^kB。两个复数累加器计算蝶形的两个输出 A' 和 B'。

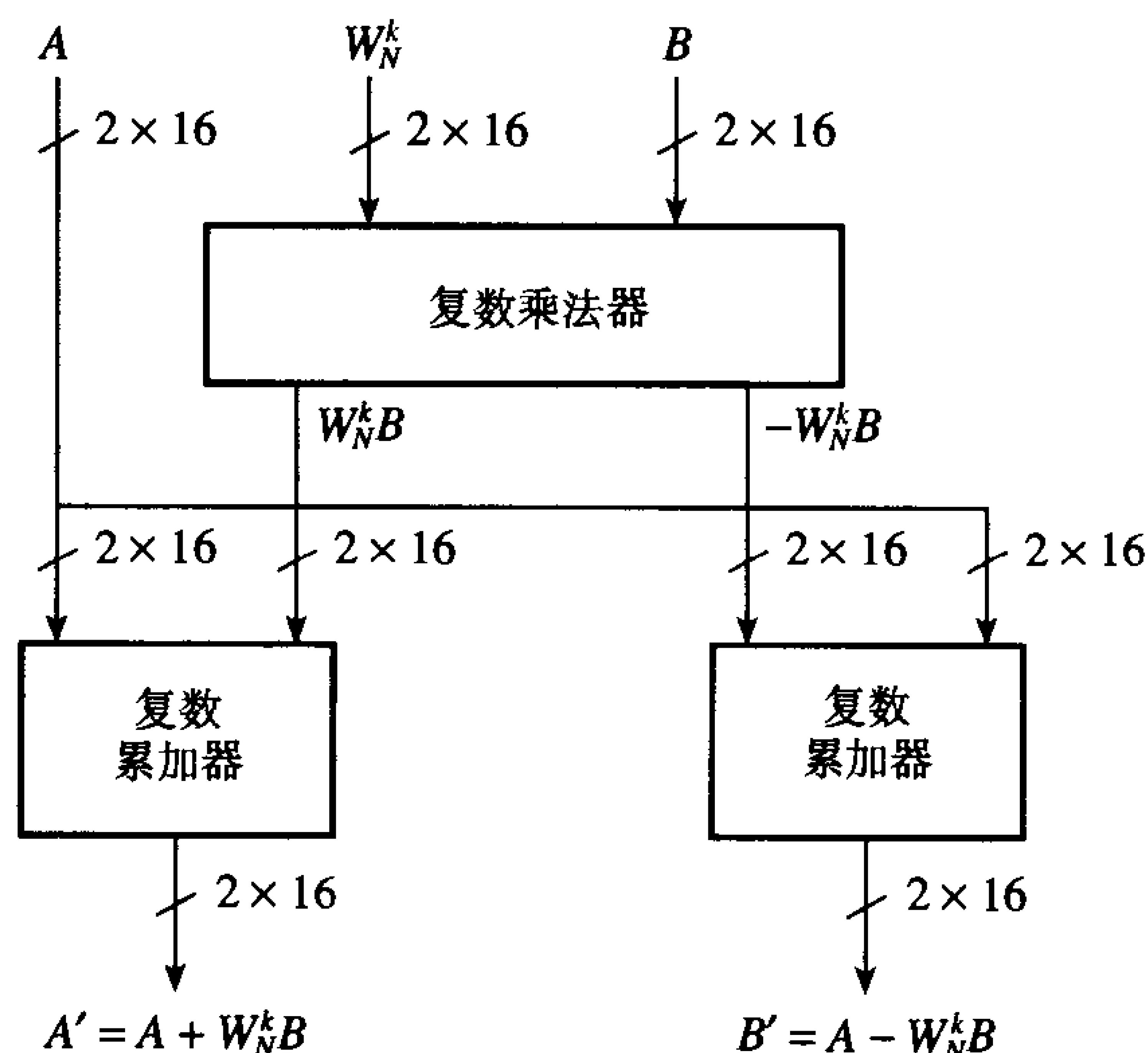


图 12.40 使用复数算术单元分立模块构造的一个硬件蝶形处理器的概念

一个 50 ns、单周期的蝶形处理器可以使用 Mitel 的复数乘法器（PDSP16112A）和一对复数累加器（PDSP16318A）实现。使用标准的实数算术单元，一个等效的蝶形处理器由四个乘法器和六个加法器组成。复数算术单元的使用显然可以减少芯片的数量，并且可能增强系统的性能。围绕蝶形处理器构造的一个硬件 FFT 处理器如图 12.41 所示。单机的高速 FFT 处理器，比如 Mitel 的 PDSP16510 器件也是可以买到的。

一个实时的、双缓冲的 FFT 配置在图 12.42 中描述。N 点 FFT 在两个缓冲器中交替执行。在缓冲器 A 执行 N 点数据的 FFT 时，缓冲器 B 填充新的数据。双缓冲允许实时连续 FFT 而不丢失数据。完成 N 点 FFT 的最大时间间隔是 $T_f = NT$(s)。

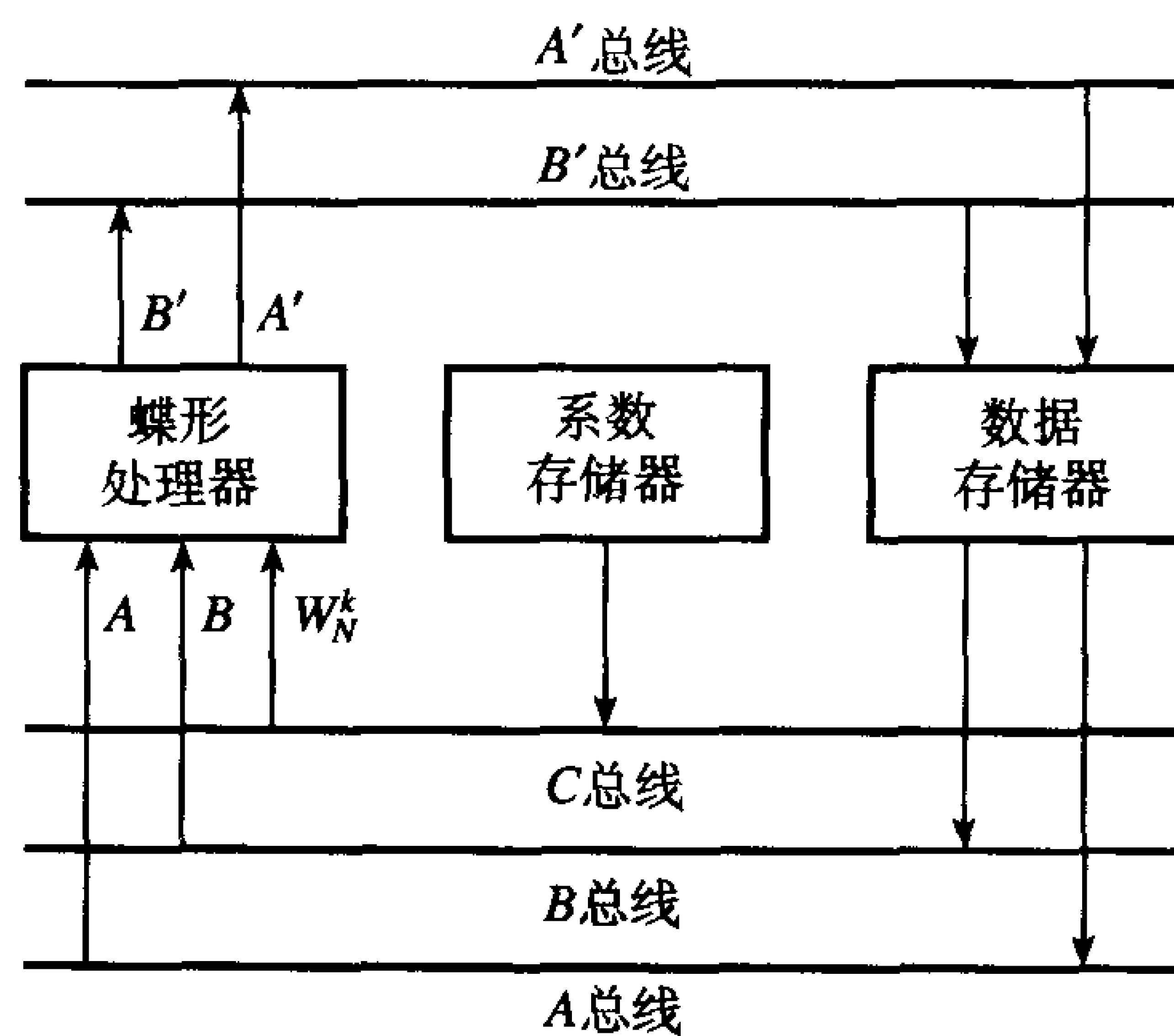


图 12.41　一个硬件 FFT 处理器的一种简化体系结构。控制器和地址产生器没有显示

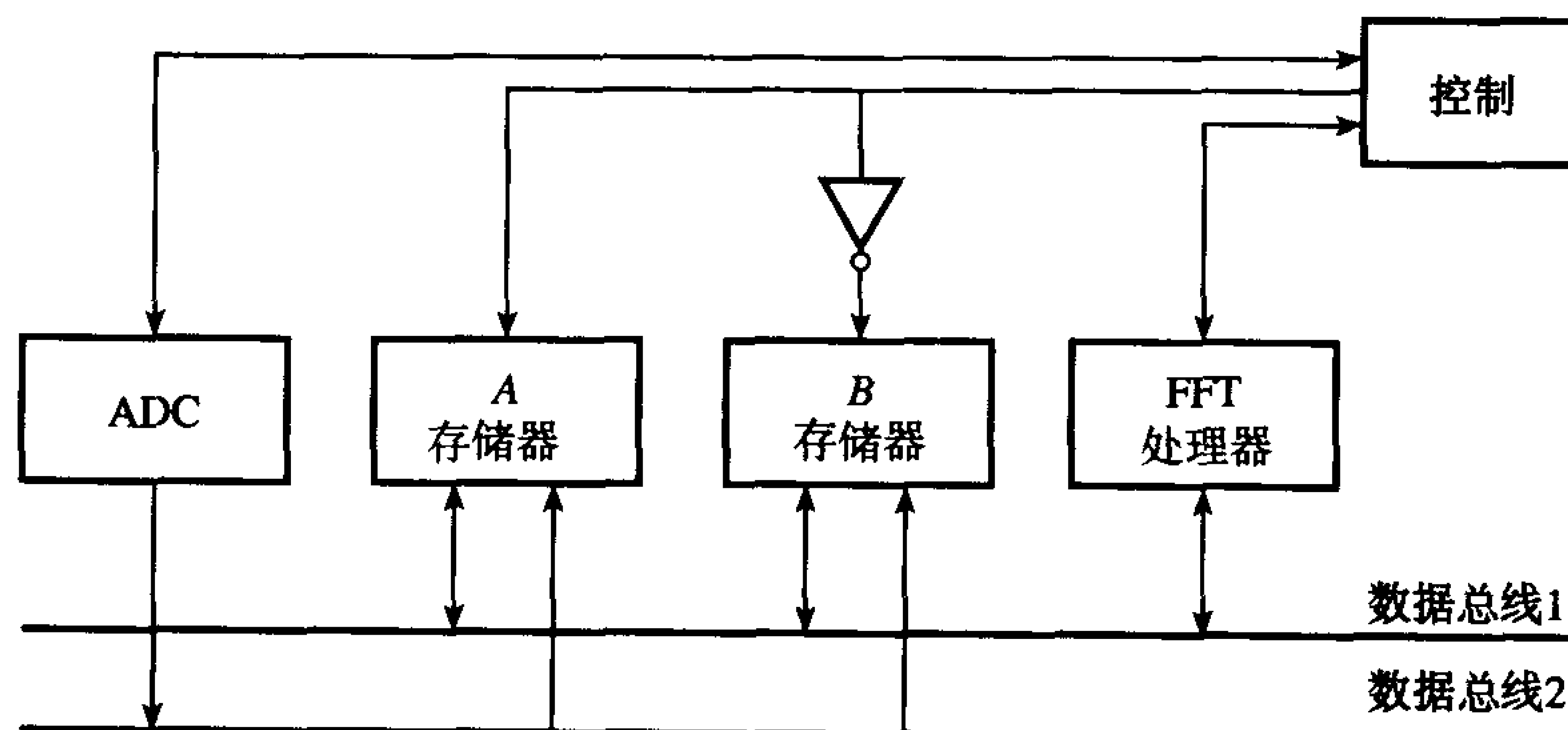


图 12.42　实时 FFT 中的双缓冲

12.7　小结

DSP 算法涉及广泛的算术操作，特别是通过 CPU 的大数据流的乘法和加法操作。这些算法在实时条件下的有效执行，要求一种与标准微处理器根本不同的硬件体系结构和指令集。在数字信号处理器中，这是通过使用哈佛体系结构、流水线技术和专用硬件（例如快速硬件乘法－累加器和移位器）的概念以及通过提供快速内部存储器和很多面向 DSP 的指令来实现的。

为了迎接多通道、计算密集型应用——比如现代远程访问服务器调制解调器、3G（第三代）移动通信和多媒体信息处理的挑战，人们又引入了新的体系结构。特别是，在最新一代 DSP 处理器中使用的超长指令字（VLIW）和静态超标量体系结构。这两种体系结构具有多数据通道和算术单元，在指令级利用并行技术来增加性能。

有两种类型的数字信号处理器：通用处理器（与标准微处理器类似，除了它们具有适用于 DSP 操作的体系结构和指令集）和专用处理器。后者用于执行专门的 DSP 算法，例如数字 FIR 滤波（算法专用的数字信号处理器），或者用于一些依赖应用的操作的有效执行（应用专用的数字信号处理器）。和通用数字信号处理器相比，专用 DSP 处理器提供了速度优势，但是缺乏灵活性。

本章详细讨论了 DSP 硬件的基本思想和 DSP 算法对 DSP 处理器体系结构的影响，并讨论了一些关键 DSP 算法的实现（使用通用数字信号处理器以及专用 DSP 处理器）来解释所涉及到的问题。

习题

12.1 给下面每个概念写出简短关键的注释，在适当的地方使用图表来解释你的答案：

- 哈佛体系结构；
- 流水线技术；
- 乘法 - 累加器；
- 特殊指令；
- 数据和程序存储器。

解释TMS320系列使用的哈佛体系结构和严格的哈佛体系结构有何不同。比较这种体系结构和标准冯·诺伊曼处理器的体系结构。

12.2 (1) 一个数字信号处理器需要一个三级流水的乘法 - 累加器。为该MAC画出一个适当配置的框图。以时序图为辅助，简要解释你的MAC是如何工作的。

(2) 假定存储器存取时间为150 ns，乘法时间为100 ns，加法时间为100 ns，每个流水阶段有5 ns开销。确定该MAC的吞吐率。注释你的答案。

(3) 该DSP系统要求实时执行下面的算法：

$$y(n) = a_0x(n) + a_1x(n-1) + a_2x(n-2) + \ldots + a_{N-1}x[n-(N-1)]$$

MAC计算每个输出样本需要花费多长时间?

12.3 M.J. Flynn在他的文章中（Flynn, 1966）将高速计算机分为下列四类：

(1) 单指令流，单数据流（SISD）;

(2) 单指令流，多数据流（SIMD）;

(3) 多指令流，单数据流（MISD）;

(4) 多指令流，多数据流（MIMD）。

这里指令流是计算机执行的程序指令序列，数据流是计算机执行指令要求的数据序列。用正当的理由确定下列每个处理器的合适类别：

- 摩托罗拉 68000;
- 摩托罗拉 DSP56000;
- 模拟器件公司 ADSP2100;
- 德州仪器 TMS320C50;
- 德州仪器 TMS320C30;
- 德州仪器 TMS320C40;
- 德州仪器 TMS320C62X;
- 模拟器件公司 TS001。

12.4 (a) 解释为什么传统衡量指标比如处理器时钟速度、MIPS和MFLOPS可能不适合于比较DSP处理器的执行性能。提出比较执行性能的一种可供选择的方法，并给出合适理由。

(b) 说出并讨论除执行速度之外的、为下列每个应用选择DSP处理器时应该考虑的四个关键因素：

(i) 高保真数字音频；

(ii) IP 电话；

(iii) 用于生物医学诊断的生理信号处理。

12.5 (a) 比较 TMS320C50、DSP56000 和 ADSP2100 定点处理器的计算性能，根据 N 点 FIR 滤波器内循环的执行速度。说明所做的任何假定。

(b) 对于 M 个二阶标准滤波器节串联组成的 N 阶 IIR 滤波器重复(a)。假定 N 是偶数。

(c) 对于在 12.16 式中定义的 LMS 自适应滤波器重复(a)。

12.6 对于下列每项技术，写出和 DSP 处理器相关的、简短的解释性注释，在适当的地方辅以草图：

(1) 环形寻址；

(2) SIMD（单输入多数据）；

(3) 超标量体系结构；

(4) 超长指令字体系结构；

(5)零开销循环。

在每个例子中，清楚地指出该技术在信号处理中的优点和缺点。

12.7 图 12.43 给出的是一个 16 点 DIT FFT 信号流图。构造等效的不变几何信号流图。对两种流图的相对优点做出评价。

12.8 一个 8 点 DIT FFT 的信号流图如图 12.32 所示，输出是搅乱的。说明自然顺序的输出可以通过倒位 $X(k)$得到。然后说明在 DIT FFT 中，通过在进行 FFT 之前搅乱输入数据序列或者在 FFT 之后归整输出，最后的输出将以正确的顺序出现。

12.9 一个 16 点 FFT 的输入数据序列以及其下标的二进制表示在表 12.8 中给出。确定倒位顺序的输入序列，然后完成这个表。

表 12.8　习题 12.9 的输入数据序列

输入序列，自然顺序	序列的二进制代码	输入序列，倒位	序列（倒位）的二进制代码
$x(0)$	0000		
$x(1)$	0001		
$x(3)$	0011		
$x(5)$	0101		
$x(6)$	0110		
$x(7)$	0111		
$x(8)$	1000		
$x(9)$	1001		
$x(10)$	1010		
$x(11)$	1011		
$x(12)$	1100		
$x(13)$	1101		
$x(14)$	1110		
$x(15)$	1111		

12.10 使用算术基本元件，为一个以二阶节串联形式实现的实时 N 阶 IIR 数字滤波器设计一个有效的专用硬件。假定 ADC/DAC 的分辨率是 12 位，系数字长是 16 位。要求 100 kHz 的抽样率。说出所做的任何假定。

12.11 (a) 为下列每一项写出和基于 DSP 的系统相关的解释性注释：

(i) 动态范围；

(ii) 定点和浮点算术。

参考在音频、通信和生物医学领域的专门应用来解释你的答案。

(b) (i) 一个基于 DSP 的系统使用 16 位字长的定点数字信号处理器。估计该系统提供的动态范围。

(ii) 如果字长是 24 位重复(i)。

(c) 如果乘法－累加器提供一个保护位以防止溢出，重复(b)。

(d) 如果使用八个保护位，重复(c)。

估计并比较上面每一项得到的附加动态范围。

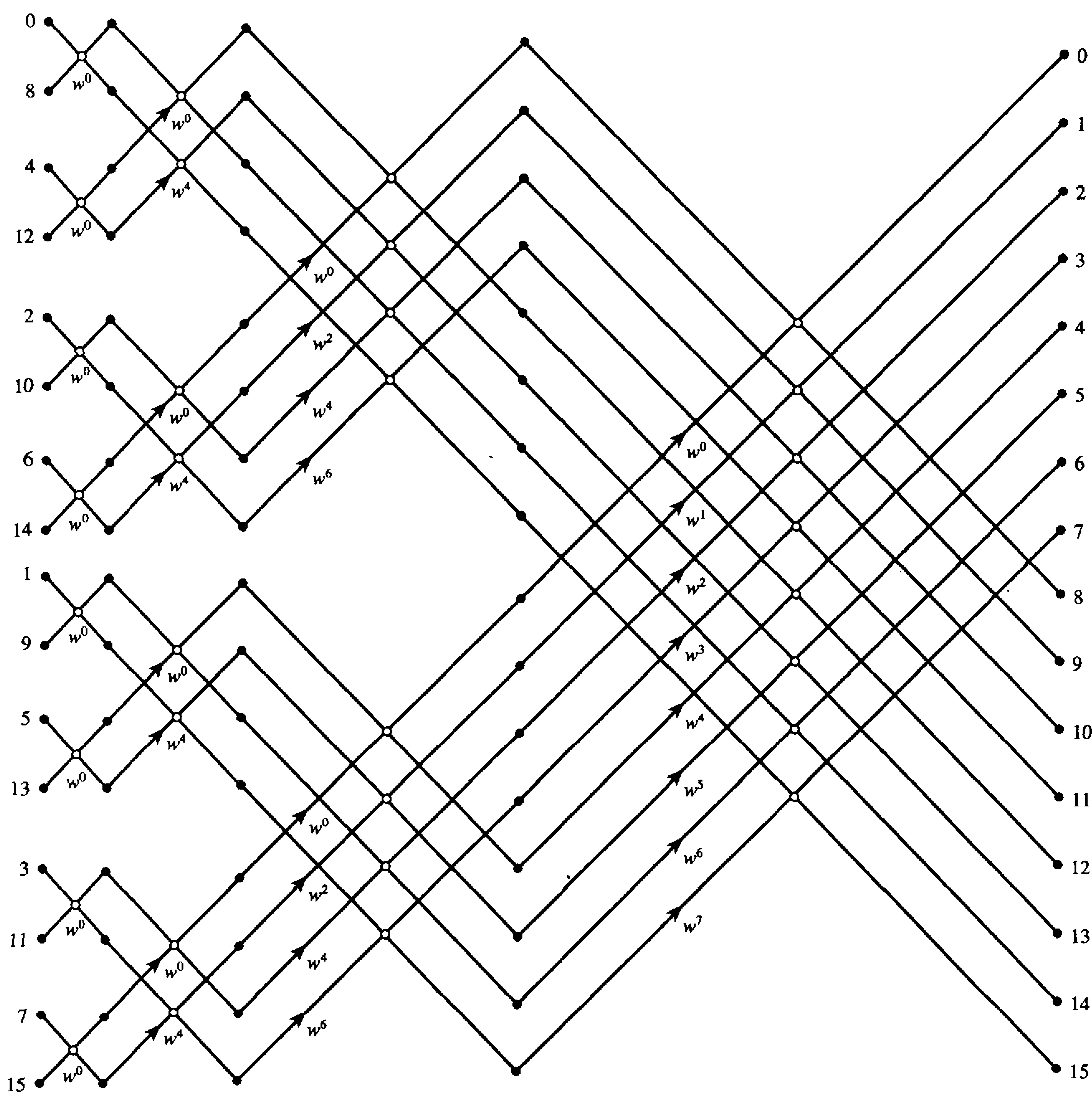


图 12.43　一个 16 点基-2 DIT FFT 流图

参考文献

Berkeley Design Technology (1999) *Buyer's Guide to DSP Processors*. Fremont CA: Berkeley Design Technology Inc. Details available at *www.BDTI.com*

Blalock G. (1997) General-purpose μPs for DSP applications: consider the trade-offs. *EDN*, 23 October, pp. 165–72.

Flynn M.J. (1966) Very high-speed computing systems. *Proc. IEEE*, **54**(12), 1901–9.

Hacker S. (1999) Static superscalar design: a new architecture for the TigerSHARC DSP processor. Analog Devices Whitepaper. *www.analog.com/publications/whitepapers/products/sharc.html*

Hayes J.P. (1998) *Computer Architecture and Organization*, 3rd edn. Boston MA: McGraw-Hill.

Hennessy J.L. and Patterson D.A. (1990) *Computer Architecture: A Quantitative Approach*. San Mateo CA: Morgan Kaufmann.

Levy M. (1998) DSP architecture directory. *EDN*, 23 April, pp. 40–110.

Levy M. (1999) DSP architecture directory. *EDN*, 15 April, pp. 67–102.

Rabiner L.R. and Gold B. (1975) *Theory and Applications of Digital Signal Processing*. Englewood Cliffs NJ: Prentice-Hall.

Texas Instruments (1988) *TMS320C2x Software Development System User's Guide*. Dallas TX: Texas Instruments.

Texas Instruments (1999) TMS320C6000 Technical Brief, Literature No. SPRU197D, Texas Instruments, Austin TX. Also available at *www.ti.com*

参考书目

Casey P.E. and Simmers L. (1986) Digital signal processing IC helps to shed new light on image processing applications. *Electronic Design*, 20 March, 135.

Chassaing R. and Horning D.W. (1990) *Digital Signal Processing with the TMS320C25*. New York: Wiley.

Cragon H. (1980) The elements of single-chip microcomputer architecture. *Computing Mag.*, **13**(10), 27–41.

Croisier A., Estaban D.J., Levilion M.E. and Rizo W. (1973) *Digital Filter for PCM Encoded Signal*. US Patent 3777130, 3 December, 1973.

Dahnoun N. (2000) *Digital Signal Processing Implementation Using the TMS320C6000 DSP Platform*. Englewood Cliffs NJ: Prentice-Hall.

De Roberts R.B. and Rifaat R. (1999) DSPs enhance flexible third-generation base-station design. *Wireless System Design*, November, *www.wsdmag.com*

DSP-architecture directory, 1997, *EDN*, 8 May, 43–107.

DSP56300 Family Manual, Motorola, *www.motorola.com/SPS/DSP/documentation*

DSP56000 Digital Signal Processor Family Manual, Motorola, Austin TX, 1995.

Fine B. and McGuire G. Considerations for selecting a DSP processor – ADSP-2101 vs TMS320C50, AN-233 Analog Devices Application Note, Digital Signal Processing Products 9-77–9-92.

Gallant J. (1990) Plug-in DSP boards. *EDN*, **35**, 142–60.

Ganesan S. (1991) A dual-DSP microprocessor system for real-time digital correlation. *Microprocessor and Microsystems*, **15**(7), 379–84.

Gore A.E. (1986) Cascadable digital signal processor. *New Electronics*, 14 October, 39–40.

Jouppi P. and Wall D.W. (1989) Available instruction-level parallelism for superscalar and superpipelined machines. In *Proc. Third Conf. on Architectural Support for Programming Languages and Operating Systems*. IEEE/ACM (April), Boston MA, pp. 272–82.

Kloker K.L. (1986) The Motorola DSP56000 Digital Signal Processor, *IEEE Micro*, December, pp. 29–48.

Kogge P.M. (1981) *The Architecture of Pipelined Computers*. New York: McGraw-Hill.

Leary K. and Morgan D. (1986) Fast and accurate analysis with LPC gives a DSP chip speech-processing power. *Electronic Design*, 17 April, 153.

Levy M. (1996) DSP-chip directory, *EDN*, March, 42–103.

Levy M. (1997) C compilers for DSPs flex their muscles. *EDN*, 7 June, 93–107.

Lin K.S. (ed.) (1988) *Digital Signal Processing Applications with the TMS320 Family*, Vol. 1. Englewood Cliffs NJ: Prentice-Hall.

Lin K.S., Frantz G.A. and Simar R. (1987) The TMS320 family of digital signal processors. *Proc. IEEE*, **75**(9), 1143–59.

McKee D. (1990) TMS32010 routine finds phase. *EDN*, **35**, 148.

Mennen P. (1991) DSP chips can produce random numbers using proven algorithm. *EDN*, **36**, 141–6.

Messer D.D. (1991) Convolutional encoding and viterbi decoding using the DSP56001. *Microprocessors and Microsystems*, **15**(1), 54–62.

Motorola (1986) *DSP56000 Digital Signal Processor User's Manual*. Motorola.

Nath N.S.M. (1999) C compilers and development tools simplify DSP assembly-language programming. *EDN*, 21 January, 103–10.

Papamichalis P. (1990) *Digital Signal Processing Applications with the TMS320 Family. Theory, Algorithms, and Implementations*, Vol. 3. Dallas TX: Texas Instruments.

Papamichalis P. and Simar R. (1988) The TMS320C30 floating-point digital signal processor. *IEEE Micro Mag.*, **8**(6), 10–28.

Roesgen J. (1986) Fast modem designs benefit from DSP chip's versatility. *Electronic Design*, 12 June.

Roesgen J. and Tung S. (1986) Moving memory off chip, DSP μP squeezes in more computational power. *Electronic Design*, 20 February, 131.

Rosen S. (1969) Electronic computers: a historical survey. *Computer Survey*, **1**(1), 7–36.

Schmalzel J., Hein D. and Ahmed N. (1980) Some pedagogical considerations of digital filter hardware implementation. *IEEE Circuits and Systems Mag.*, **2**(1), 4–13.

So J. (1983) TMS 320 – step forward in digital signal processing. *Microprocessors and Microsystems*, **7**(10), 451–60.

Stokes J. and Sohie G.R.L. (1991) Implementation of PID controllers on the Motorola DSP56000/DSP56001. Part 1. *Microprocessors and Microsystems*, **15**(6), 321–31.

Stokes J. and Sohie G.R.L. (1991) Implementation of PID controllers on the Motorola DSP56000/DSP56001. Part 2. *Microprocessors and Microsystems*, **15**(7), 385–92.

Texas Instruments (1989) *Second-Generation TMS320 User's Guide*. Dallas, TX: Texas Instruments.

TMS320C5x User's Guide, Texas Instruments, 1995.

TMS320C54x DSP Reference Set, Volume 4: Applications Guide. Texas Instruments, October 1996. *www.ti.com*

Tomarakos J. and Ledger D. (1998) Using the Low-cost, High Performance ADSP-21065L Digital Signal Processor for Digital Audio Applications. Analog Devices DSP Application. Details are available at *www.analog.com*

Zolzer U. (1997) *Digital Audio Signal Processing*. Wiley.

其他有用的 Web 地址

Special purpose hardware, e.g. MT9300 multi-channel voice echo canceller, PDSP16256 programmable FIR filters. Mitel Semiconductor, *www.mitelsemi.com*

Audio codecs and processors, e.g. CS4228 24 bit 96 kHz surround sound codec, digital audio sample rate converters. Cirrus, *www.cirrus.com*

附录

12A　实时信号处理的 TMS320 汇编语言程序和不变几何基-2 FFT 的 C 语言程序

下列 TMS320C10/C25 程序可以在配套的指导手册包含的 CD 中找到（细节见前言）。

(1) 基于 TMS320C10 的 FIR 数字陷波滤波器；
(2) FIR 数字带通滤波器的 TMS320C10 实现；
(3) 基于 TMS320C25 的 FIR 数字陷波滤波器；
(4) TMS320C10 四阶数字 IIR 滤波器，二阶节串联形式实现；
(5) TMS320C25 四阶数字 IIR 滤波器，二阶节串联形式实现；
(6) TMS320C25 四阶数字 IIR 滤波器，二阶节并联形式实现；
(7) 不变几何基-2 FFT 的 C 语言程序；
(8) 基于 TMS320C25 的基-2 FFT 算法；
(9) TMS320C25 自适应滤波器。

因为缺少空间，只有程序(2)和程序(5)在此附录中列出。

程序 12A.1　基于 TMS320C10 实现的 FIR 数字带通滤波器

```
METAi Assembler 4.00 ©1988 Crash Barrier Thu Nov 19 00:37:40 1992
Page 1 Assembler
                                        targbpf.asm

00000000                     1  c:\metal\32010.tab/
                             2
00000000                     3        .ctrl              27, 15
00000000                     4        SEGMENT            word at 0000 'ram'
00000000                     5        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
00000000                     6        ;                                                        ;
00000000                     7        ;    FIR BANDPASS FILTER                                 ;
00000000                     6        ;                                                        ;
00000000                     9        ;    Filter specification:                               ;
00000000                    10        ;                                                        ;
00000000                    11        ;    filter type                     bandpass filter     ;
00000000                    12        ;    sampling frequency              15 kHz              ;
00000000                    13        ;    passband                        900–1100 Hz         ;
00000000                    14        ;    transition width                450 Hz              ;
00000000                    15        ;    passband ripple                 <0.87 dB            ;
00000000                    16        ;    stopband attenuation            >30 dB              ;
00000000                    17        ;    filter length                   41                  ;
00000000                    18        ;                                                        ;
00000000                    19        ;    Hardware: TMS320C10 Target board with 8-bit ADC/DAC ;
00000000                    20        ;                                                        ;
00000000                    21        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
00000000                    22        ;
                            23
00000000     F900002B       24        B       START
                            25
00000028                    26  NM1   EQU     40          ;N–1
00000000                    27  XN    EQU     0           ;CURRENT I/P SAMPLE
```

```
00000028                          28  XNM1   EQU   NM1
00000029                          29  H0     EQU   NM1+1
00000051                          30  HNM1   EQU   H0+NM1
0000007B                          31  YN     EQU   123
0000007C                          32  ONE    EQU   124
00000000                          33  PA0    EQU   0        ;address of I/O for D/A IN TARGET BOARD
00000001                          34  PA1    EQU   1
00000002                          35  PA2    EQU   2
00000003                          36  PA3    EQU   3
00000002                          37  COEFF  EQU   2        ;START ADDRESS OF COEFFS.
00000000                          38  R0     EQU   0
00000001                          39  R1     EQU   1
00000002                          40  ;
00000002                          41  ;TABLE OF COEFFS. THESE ARE INITIALLY
00000002                          42  ;STORED IN PROGRAM MEMORY.
                                  43
                                  44
00000002  FE09FFFEFF5B019F        45
00000002  02B3038E03D9            45         DC.W  -503,-2,-165,415,691,910,985
00000009  035001 D9FF97FCE8       46
00000009  FA57F881                46         DC.W  846,473,-105,-792,-1449,-1919
0000000F  F7EAF8DDFB52FEE8        47
0000000F  02F406A8                47         DC.W  -2070,-1627,-1198,-280,756,1704
00000015  093F0A2E093F06A8        48
00000015  02F4FEE8                48         DC.W  2367,2606,2387,1704,756,-280
0000001B  FB52F8DDF7EAF881        49
0000001B  FA57                    49         DC.W  -1198,-1827,-2070,-1919,-1449
00000020  FCE8FF9701D90350        50
00000020  03D9038E02B3            50         DC.W  -792,-105,473,848,985,910,691
00000027  019F00A5FFFEFE09        51         DC.W  415,165,-2,-503
```

METAi Assembler 4.00 © 1988 Crash Barrier Thu Nov 19 00:37:40 1992
Page 2 Assembler

targbpf.asm

```
                       52
                       53
0000002B               54  ;================ START OF MAIN PROGRAM ================
0000002B               55  ;
0000002B               56  ;INITIALIZATION
0000002B               57  ;
0000002B  7E01         58  START  LACK  1
0000002C  507C         59         SACL  ONE
0000002D  6E00         60         LDPK  0         ;POINT TO PAGE ZERO OF DATA
                                                   ;MEMORY
0000002E               61  ;
0000002E               62  ;TRANSFER COEFFICIENTS TO DATA MEMORY FROM PROGRAM
                           ;MEMORY IN EPROM SPACE
0000002E               63  ;
0000002E  7E02         64         LACK  COEFF     ;LOAD COEFF ADDRESS INTO
                                                   ;ACCUMULATOR
0000002F  7028         65         LARK  AR0,NM1   ;NO OF COEFF INTO AUXILIARY
                                                   ;REGISTER 0
00000030  7129         66         LARK  AR1,H0    ;AND DATA MEMORY ADDRESS OF
                                                   ;COEFFICIENTS INTO AR1
00000031  6881         67  LOAD   LARP  R1        ;SELECT AR1 AND BEGIN TO TRANSFER
                                                   ;COEFF.
00000032  67A0         68         TBLR  *+,R0     ;INTO DATA MEMORY, THEN INCREMENT
                                                   ;THE CONTENTS OF AR1
00000033  007C         69         ADD   ONE       ;INCREMENT THE ACCUMULATOR
00000034  F4000031     70         BANZ  LOAD      ;DEC AR0, AND BRANCH IF NOT ZERO
00000036               71  ;
00000036               72  ;WAIT FOR NEW INPUT SAMPLE
00000036               73  ;
```

```
00000036    6880        74          LARP   R0
00000037    F600003B    75  WAIT    BIOZ   NXTPT    ;SEE IF SAMPLE IS RDY
00000039    F9000037    76          B      WAIT     ;IF NOT GO WAIT
                        77
0000003B    4000        78  NXTPT   IN     XN,PA0   ;IF READY THEN READ SAMPLE . . . was
                                                    ;PA2 for EVM
0000003C                79  ;
0000003C                80  ;CALCULATE FILTER OUTPUT IN YN AND OUTPUT TO DAC
0000003C                81  ;
0000003C    7028        82  skip    LARK   AR0,XNM1
0000003D    7151        83          LARK   AR1,HNM1
0000003E    7F89        84          ZAC
                        85
0000003F    6A91        86          LT     *-,R1    ;LOAD XN(N-1) SAMPLE
00000040    6D90        87          MPY    *-,R0    ;COMPUTE H(N-1)*XN(N-1)
00000041    6B81        88  LOOP    LTD    *,R1     ;COMPUTE SIG[H(K)*X(N-K)]
00000042    6D90        89          MPY    *-,R0
00000043    F4000041    90          BANZ   LOOP
00000045    7F8F        91          APAC            ;ADD H(N-1)*X(N-1)
00000046    597B        92          SACH   YN,1     ;OUTPUT SAMPLE
00000047    487B        93          OUT    YN,PA0   ;was PA3 for EVM
                        94
00000048    F6000048    95  onhi    BIOZ   onhi     ;wait here until BIO line goes high before
                                                    ;going for next
                        96
0000004A    F9000037    97          B      WAIT
                        98
0000004C                99          end
No errors on assembly of 'targbpf.asm'
```

程序 12A.2　TMS320C25 四阶数字 IIR 滤波器，二阶节串联形式实现

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Fourth order Elliptic filter, connected                   ;
; as a cascade of 2 biquad canonic sections                 ;
; Manny Ifeachor, Jan., 1992                                ;
;                                                           ;
;       FILTER SPECIFICATIONS:                              ;
;                                                           ;
;       Filter Type                 lowpass                 ;
;       Sampling frequency          15 kHz                  ;
;       Passband                    0–3 kHz                 ;
;       transition width            450 Hz                  ;
;       Passband ripple             0.5 dB                  ;
;       Stopband attenuation        45 dB                   ;
;                                                           ;
;       Hardware: TMS320C25 SWDS with AIB                   ;
;       1-bit ADC/DAC (filter B)                            ;
; ----------------------------------------------------------;
;
XN        .set    0
YN1       .set    1
W1N       .set    2
W1NM1     .set    3
W1NM2     .set    4
YN2       .set    5
W2N       .set    6
W2NM1     .set    7
W2NM2     .set    8
SF1       .set    9
A01       .set    10
```

```
A11     .set    11
A21     .set    12
B11     .set    13
B21     .set    14
A02     .set    15
A12     .set    16
A22     .set    17
B12     .set    18
B22     .set    19
SF2     .set    20
ONE     .set    21
RATED   .set    22
MODED   .set    23
WONE    .set    24
TEMP    .set    25
PBM1    .set    0300 h
;
        .sect   "IRUPTS"
START   B       INIT
;
        .text
COEFFS  .word    13217          ; SF1 = 0.4033627 IN Q15 FORMAT
                                ;(SCALE FACTOR)
;
        .word    4296           ;a01=0.1311136
        .word    7087           ;a11=0.2162924
        .word    4296           ;a21=0.1311136
        .word    27175          ;-b11=0.829328
        .word   -10061          ;-b21=-0.307046
;
        .word    32767          ;a02=0.9999695 (largest + ve number)
        .word    22142          ;a12=0.675718
        .word    32767          ;a22=0.9999695
        .word    16251          ;-b12=0.495935
        .word   -24965          ;-b22=-0.761864
;
        .word    29769          ;SF2=0.90847
;
MODEP   .word   0Ah
RATEP   .word   0299h
;
*
**      INITIALIZE THE AIB **
*
INIT    LDPK    6
        SSXM
        LACK    MODEP
        TBLR    MODED
        OUT     MODED,PA0
        LACK    RATEP           ;SET UP AIB SAMPLING FREQ
        TBLR    RATED
        OUT     RATED,PA1
        OUT     RATED,PA3       ;LET GO AIB
*
**      TRANSFER COEFFS FROM PROG MEMORY TO DATA MEMORY
*
        LARP    AR0
        LRLK    AR0,PMB1+SF1
        RPTK    11
        BLKP    COEFFS,*+
*
**      INTIALIZE DMA FOR INTERNAL NODE DATA
```

```
*
INITWN  ZAC
        SACL  W1N
        SACL  W1NM1
        SACL  W1NM2
        SACL  W2N
        SACL  W2NM1
        SACL  W2NM2
*
**      WAIT FOR NEW DATA SAMPLE TO BE READY
*
RDATA   BIOZ  NXTPT          ;FETCH THE NEW SAMPLE
        B     RDATA
NXTPT   IN    XN,PA2
        LT    XN
*
**      START OF FILTER BLOCK 1
*
BLOCK1  MPY   SF1            ;SCALE INPUT DATA SAMPLE: SF*X(N)
        PAC
        LT    W1NM1          ;LOAD T-REGISTER WITH W(N-1)
        MPY   B11            ;B11W1(N-1)
        LTA   W1NM2          ;SFX(N)+B11W1(N-1)
        MPY   B21            ;B21W1(N-2)
        APAC                 ;SFX(N)+B11W1(N-1)+B21W1(N-2)
        SACH  W1N
        ZAC
        MPY   A21            ;A21W1(N-2)
        LTD   W1NM1          ;SUM = A21W(N-2); W1(N-2)=W1(N-1)
        MPY   A11            ;A11W1(N-1)
        LTD   W1N            ;SUM=A21W1(N-2) + A11W1(N-1); W1(N-1)=W1(N)
        MPY   A01            ;A01W1(N)
*
**      START OF FILTER BLOCK 2
*
BLOCK2  LTA   W2NM1          ;Y1(N)=A21W(N-2)+A11W1(N-1)+A01W1(N)
        MPY   B12            ;B12W2(N-1)
        LTA   W2NM2          ;Y1(N)+B12W2(N-1)
        MPY   B22            ;B22*W2(N-2)
        APAC                 ;Y1(N)+B12*W2(N-1)+B22*W2(N-2)
;                            ;SUM=Y1(N)+B12*W2(N-1)+2*B22*W2(N-2)
        SACH  W2N            ;STORE HIGH 16 BIT WORD OF SUM IN W2(N)
        MPY   A22            ;A22*W2(N-2)
        ZAC                  ;SUM=SUM+A22*W2(N-2)
        LTD   W2NM1          ;W2(N-2)=W2(N-1); SUM = A22*W2(N-2)
        MPY   A12            ;A12*W2(N-1)
        APAC                 ;SUM=A22*W2(N-2)+A12*W2(N-1)
        LTD   W2N            ;W2(N-1)=W2(N)
        MPY   A02            ;A02*W2(N)
        APAC
        SACH  YN2            ;
*
**      SCALE OUTPUT SAMPLE AND SEND IT TO DAC
*
        LT    YN2            ;SCALE OUTPUT BACKUP
        MPY   SF2            ;SF2*YN2
        PAC                  ;
        APAC
        APAC
        APAC
        APAC
```

```
            APAC
            APAC
            APAC
            APAC
            APAC
            APAC
            APAC
            APAC
            APAC
            APAC
            APAC
            APAC
            APAC
            APAC
            APAC
            SACH    YN2             ;21*SF2*YN2
;
            OUT     YN2,PA2
            B       RDATA
;
            END
;@\\\000300AB1800001100AB18;

/* This is the link command file */
MEMORY
{                                                           /* Program Memory */
    PAGE 0: VECTORS:origin=0h, length=01Fh
            CODE:origin=20h, length=0F90h
                                                            /* Data Memory */
    PAGE 1: RAMB2:origin=60h, length=020h
            RAMB0:origin=200h, length=0FFh
            RAMB1:origin=300h, length=0FFh
}
SECTIONS
{   IRUPTS      :{}     > VECTORS   PAGE 0
    .text       :{}     > CODE      PAGE 0
    .data       :{}     > RAMB2     PAGE 1
    .bss        :{}     > RAMB0     PAGE 1
}
```

第13章　定点DSP系统的有限字长效应分析

本章的目的是了解在实际的DSP系统中，由于量化和使用有限字长算术单元来实现DSP操作所带来的误差。还要讨论误差对信号质量带来的影响和如何克服它们。读者通过对本章的这一问题的理解，有助于设计性能可预期的实用DSP系统。

本章的重点是定点DSP系统，因为它们正被广泛地使用。

13.1　引言

在大多数情况下，DSP设计问题的最终目标是实现一个DSP功能，例如在一个数字处理器中的滤波或FFT。在实际中，一定个数的比特位用来表示变量和进行算术操作。现代DSP处理器的典型字长是16位（例如TMS320C54）、24位（例如DSP56300）和32位（例如ADSP-21065）。使用有限字长所带来的误差会影响DSP系统的性能。在实现一个DSP功能之前，设计者必须确定有限字长效应产生的误差导致性能下降的程度，如果必要，可以找到相应的解决方案。

DSP的主要误差有

(1) ADC量化误差——这是由于用一个有限长度的比特数来表示输入数据而产生的。

(2) 系数量化误差——这是由于用一个有限长度的比特数来表示系数或DSP参数而产生的。例如在二级滤波器设计中，系数a_k和b_k通常具有很高的精度；但在实际的DSP处理器中，它们必须根据处理器的字长而被量化。

(3) 溢出误差——这是由于两个很大的同符号数相加，其结果超出了容许的字长所产生的。

(4) 舍入误差——这是当乘法的结果被舍入(或截断)到最近的离散值或容许的字长时产生的。

信号处理过程中产生的误差取决于多个因素，包括使用的算术类型、输入信号的质量、DSP功能的类型和实现的算法。在讨论中，我们主要采用IIR数字滤波器为参照来研究有限字长对DSP系统性能的影响，因为它集中体现了在实践中遇到的大多数问题。

13.2　DSP算术

我们在前面的章节看到，DSP中基本的操作是乘法、加法和延迟（或位移）。例如，在数字FIR中滤波系数$h(k)$（$k = 0, 1, \ldots, N-1$）与输入数据抽样$x(n)$（$n = 0, 1, \ldots$）相乘，乘积相加的结果如下：

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k) = h(0)x(n) + h(1)x(n-1) + \ldots + h(N-1)x[n-(N-1)] \quad (13.1)$$

在实际中，DSP包含的算术操作（如上面所指出的）经常是用定点或浮点算术实现的（Rabiner and Gold, 1975）。有时也使用其他类型的算术，如块浮点算术，它力图结合上面两种算术的优点。定点算术在DSP工作中使用得最广泛，因为它的实现更快、更便宜，但同时也限制了可以表示的数的范围。当加法结果超出了容许的数范围（例如IIR滤波器中大幅极限环和高阶FFT系统中的过载）

时，产生的溢出导致最终结果是可疑的。为了防止算术操作结果超出容许数的范围，对操作数进行了伸缩变换。这种伸缩变换降低了DSP系统的性能，因为它减小了可达到的信噪比。

当变量或系统系数的幅度在一个很宽的范围内变化时，浮点算术更受青睐（Flores，1963）。它允许一个更宽的动态范围，实质上消除了溢出问题。此外，浮点处理能够简化程序。DSP算法在大的机器上开发，例如在个人计算机或大型主机使用较高级的编程语言，只对核心算法进行微小的改动就能直接在DSP硬件上使用。然而，浮点算术更为昂贵，速度较慢，尽管目前带有一个内建的浮点处理器（例如德州仪器的TMS320C30）的高速数字信号处理器已经比较普及。有效的浮点软件例程也出现在公开的文献中（Texas Instruments, 1986）。因此定点与浮点的价格和速度差别显著地减小了。

DSP技术逐渐出现在同时具有更宽动态范围和高精度的应用场合中。具有大字长（24位）的定点数字信号处理器可以满足这些需求，但浮点处理能提供一个满足这些应用的更简单和自然的方法。

在需要大动态范围和高精度的应用领域中，许多情况下希望使用浮点算术。一个例子是在数字音频信号中使用数字滤波器进行实时参数化均衡。当使用者在音频带中调谐或调整均衡参数时，滤波器的系数值变化很大。对于特定的信号处理任务（例如雷达、声呐、地震学和生物医学中的频谱分析），经常需要解析大动态范围信号中很微小的分量。这些应用同时需要宽动态范围和高精度。其他应用例子包括高分辨图像工作站和通用工程计算。在这些例子中，希望的最大动态范围和精度需求在表13.1列出（Weitek, 1984）。

表13.1 动态范围和精度需求

	动态范围（位）	精度（位）
噪声消除	32	20
雷达处理	32	20
播放质量画面处理	20	20
图像处理	30	20
医用频谱分析	20	20
地震数据处理	70	20

关键点在于，影响DSP系统性能的主要因素是使用的算术类型。在下面几节中，我们将讨论两类算术（即定点和浮点）的基本概念。

13.2.1 定点运算

13.2.1.1 定点表示

在DSP中，变量经常需要用定点数的补码来表示，也即2的补码：参见表13.2中的例子。在这种表示中，二进制数在MSB（最高位）的右边，同时它也是符号位：每个数的范围从 -1 到 $1-2^{-(B-1)}$，其中 B 是表示这个数所用的位数。在DSP中一个通常的表示称为Q15格式，它使用了16位（1个符号位和15个小数位）：

0110 0000 0000 0000

 | 二进制数

正数的补码就是二进制数的自然形式，参见表13.2。负数则将相应正数补码的所有比特数取反再加上1 LSB。例如 $-3/8$ 的补码是从3/8（即0011）得到的，即 $1100+0001=1101$。

当DSP系统的输入是从一个ADC（模/数转换器）得到时，进入数字信号处理器的数据可能是偏移二进制格式；如果DSP系统的输出送到一个DAC（数/模转换器），那么也可能需要将其转化成偏移二进制。从偏移二进制转变成2的补码表示是非常简单的，将偏移二进制码的MSB取补

即可。例如，表 13.2 中偏移二进制码 1111，即 7/8，将 MSB 取补很容易得到 2 的补码(0111)。实际上，DSP 芯片总线经常比 ADC 的分辨率要宽。这种情形下，转换成 2 的补码以后，符号位被扩充并填满剩余的左边空位。例如，码(1111 1101)是 2 的补码，经过符号扩展后变成(1111 1111 1111 1101)。

表 13.2　字长 4 位系统中 2 的补码与偏移二进制数的比较

数	十进制小数	2 的补码	偏移二进制码
7	7/8	0111	1111
6	6/8	0110	1110
5	5/8	0101	1101
4	4/8	0100	1100
3	3/8	0011	1011
2	2/8	0010	1010
1	1/8	0001	1001
0	0	0000	1000
−1	−1/8	1111	0111
−2	−2/8	1110	0110
−3	−3/8	1101	0101
−4	−4/8	1100	0100
−5	−5/8	1011	0011
−6	−6/8	1010	0010
−7	−7/8	1001	0001
−8	−1	1000	0000

在定点制和 2 的补码表示中，如果每个数用 B 位来表示，则最大可表示 2^B 个不同的数，最接近数相隔大约 2^{-B}。知道每个数能够达到的十进制表示精度是非常有用的。

考虑一个十进制小数 X，包含有 d 个数字，则它的精度是 $\pm 0.5 \times 10^{-d}$。如果我们用有 B 位的二进制数来表示同样的数，则它的精度变成 $\pm 0.5 \times 2^{-B}$。为了使两种表示法保持同样的精度，需要

$$0.5 \times 10^{-d} = 0.5 \times 2^{-B}, \text{即} \quad B = d \log_2 10 \simeq 3.3d \tag{13.2}$$

例如，假设十进制数 0.234 56 被表示成二进制：那么我们需要 $3.3 \times 5 = 17$ 位才能达到同样的精度。表 13.3 总结了一个二进制系统的位数与具有相同精度的十进制位数的关系。

表 13.3　位数与精度（用十进制数字表示）之间的关系

位数	精度（十进制位数）
7	2.1
8	2.4
10	3
12	3.6
14	4.2
15	4.5
16	4.8
18	5.4
20	6.1
23	7.0
24	7.3
64	19.4

例 13.1　将一个十进制数 0.956 24 表示为

(1) 一个 Q3 的数，及

(2) 一个 Q4 的数

比较两种情况下的误差。

(3) 估计表示上面十进制数所需要的位数，且保持同样的精度。

解:

(1) 一个Q3数是一个2的补码，它有1个符号位和3个小数位。将十进制数转变成Q3格式，我们只需简单的将它乘以2^3，再把乘积舍入到最近的容许整数：$0.956\,24 \times 2^3 = 7.649\,92$，将它舍入成7 = 0111（最大容许数）。

(2) 在这种情况下，我们用1个符号位和4个小数位来表示这个数：$0.956\,24 \times 2^4 = 15.299\,84$，再舍入成15 = 01111。

在(1)和(2)中数的表示误差分别是：0.649 92/8 = 0.081 24和0.299 84/16 = 0.018 74。这种数的表示误差通常称为系数量化误差。

(3) 根据13.2式，我们需要$3.3 \times 5 = 16.5$位$\simeq$17位。

13.2.1.2 定点乘法

在定点制的乘法中，利用了两个小数的乘积依然是小数、两个整数的乘积依然是整数的特性。我们下面用一个例子来说明这一点。

例 13.2 应用2的补码的乘法计算0.5625的平方。假定操作数是Q4格式。

解:

```
        0 1001 = 0.5625
        0 1001 = 0.5625
 0 0000
   0100 1
     000 00
       00 000
         0 1001
00 0101 0001
 ↑          ↖
二进制点        当乘积被量化时这个位丢失了
```

向左移位以去除多余的符号位，然后舍入得到最后的答案：0 0101 = $0.25+2^{-4}$ = 0.3125，而不是0.316 406 25。

从例13.2中可以看出，乘法产生了一个多余的符号位，且两个5位数的乘积有10位长，需要将结果截断或舍入以节省内存。一般来说，两个B位数的乘积有$2B$长。例如，10位结果被向左移位以去除多余的符号位，再舍入成0.0101。可以对乘法结果进行舍入（或截断）操作是补码小数运算的一个主要优点，因为它意味着补码乘法运算不会出现溢出现象。然而，这种舍入（或截断）给信号带来了误差，有可能给存在反馈的DSP系统带来不稳定或一些不希望的边际效应。

13.2.1.3 定点加法

两个定点小数的加法比乘法困难得多。这是因为被加的操作数必须具有同样的Q格式，而且还要小心可能发生的溢出。

例 13.3 应用2的补码的加法计算下面两个数的和：0001 1001和0110 1101 0111 1101。

解:

操作数首先被表示成同样的Q格式再相加：

0110 1101 0111 1101
0001 1001 0000 0000
1000 0110 0111 1101
| 溢出

一种解决溢出的方法是将结果向右移1位，再设置一个溢出标志。这样结果变成

1 0100 0011 0011 1110
↑
溢出标志

另一种方法是用双精度数来表示结果，或为溢出留出足够的空间。

例13.4　假定有一4位的寄存器（1符号位和3数据位），计算下面的加法：
(1) –0.25和0.75；
(2) 0.5，0.75，–0.5。

解：
根据表13.2，$0.25_{10} = 0.010_2$，$0.5_{10} = 0.100_2$，以及$0.75_{10} = 0.110_2$。

(1)　0.75　　0.110
　–0.25　　1.110
　0.50　　10.100
　结果是：0.100

(2)　0.5　　0.100
　+0.75　　0.110
　1.25　　1.010←部分和
　–0.50　　1.100
　0.75　　10.110←最终和
　结果是：0.110

13.2.2　浮点运算

13.2.2.1　浮点表示

一个二进制浮点数X表示为两个带符号数的乘积，尾数为M，指数为E：

$$X = M \cdot 2^E \tag{13.3}$$

其中2是二进制的基。

指数决定了能够表示数的范围，尾数则决定数的精度。例如，如果指数和尾数分别用8和16位来表示，这时可表示浮点数的范围从0.5×2^{-128}到$1-(2^{-15}) \times 2^{128}$。在用来表示尾数的16位中，其中一位是符号位。另外，由于舍入效应，最低位的精度难以保证。因此浮点数的精度为$1/2^{14}(0.61 \times 10^4)$，即大约相当于十进制的4位。

13.2.2.2　IEEE浮点格式

一种最广泛应用的二进制浮点系统是IEEE 754标准（Patterson and Hennessy, 1990; IEEE, 1985）。该格式的单精度数表示如图13.1。在这个例子中，浮点（FP）数的指数项被正则化到$0<E<255$的范围内。

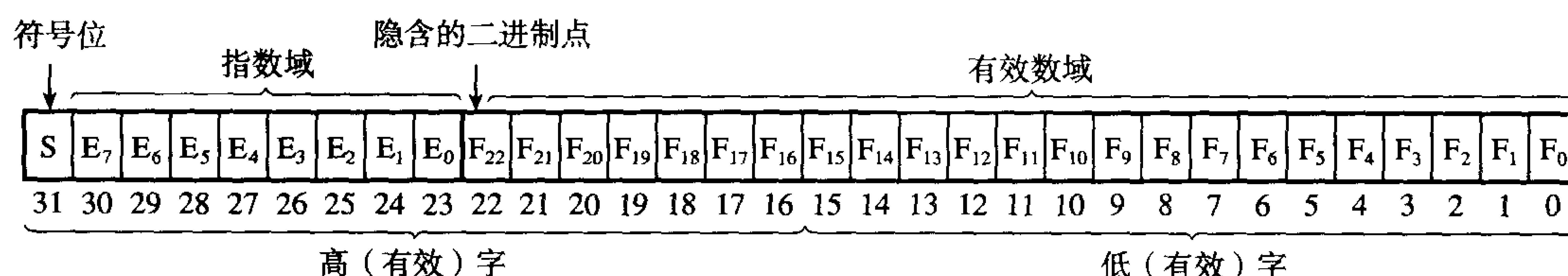


图 13.1　浮点表示（IEEE 单精度）

一个正则化 IEEE 浮点数的等效十进制数 X 由下式给出：

$$X = (-1)^s(1 \cdot F)2^{E-127}$$

其中

- F 是尾数，采用补码小数制，用 0 到 22 位来表示；
- $E-127$ 是指数；
- $s = 0$ 为正数，$s = 1$ 为负数。

IEEE 浮点格式的两个重要特征是假定尾数的前面是 1，指数是偏置（127）的。

例 13.5

(1) 对于浮点数

符号
↘0 1000 0011.1100 . . . 00000
|← 指数 →|← 尾数 →|

指数是 1000 0011 = 131，尾数是 0.1100...= 0.75，$s = 0$。因此 $X = (1.75)2^{131-127} = 1 \times 1.75 \times 2^4$。

(2) 对于浮点数

符号
↘1 0000 1111.0110 . . . 00000
|← 指数 →|← 尾数 →|

指数是 0000 1111 = 15，尾数是 0.0110...= 0.375，$s = 1$。因此 $X = (1.375)2^{15-127} = -1 \times 1.375 \times 2^{-112}$。

13.2.2.3 浮点加法和乘法

如果 X_1 和 X_2 是两个被加的浮点数，其中 $X_1 = M_1 \times 2^{E_1}$，$X_2 = M_2 \times 2^{E_2}$，则它们的和 X 为

$$X = M \times 2^E$$

其中

$$M = M_1 + M_2 \times 2^{E_1-E_2};\ E = E_1\ \text{假设}\ X_1 > X_2 \tag{13.4}$$

在两个浮点数相加以前，必须使它们的指数项相等。这称为校准，即将较小操作数的尾数右移以增加其指数，直到它与较大操作数的指数相等。

如果 X_1 和 X_2 是两个完全正则化的浮点数，其中

$$X_1 = M_1 \times 2^{E_1}, X_2 = M_2 \times 2^{E_2} \tag{13.5}$$

则它们的乘积 X 为

$$X = M \times 2^E$$

其中

$$M = M_1 \times M_2, E = E_1 + E_2$$

即尾数相乘而指数相加。由于 M_1 和 M_2 都是正则化的，因此它们的乘积 M 在 $0.25<M<1$ 的范围内。所以乘积 M 不会溢出，但可能没有很好地正则化（尾数下溢）。

例 13.6

(1) 求两个数 A 和 B 的和，其中 $A = 9.985 \times 10^4$，$B = 5.6756 \times 10^2$。

(2) 求两个数 A 和 B 的积，其中 $A = 2.75 \times 10^{-16}$，$B = 4.5 \times 10^{10}$。

解：

(1) 首先，比较两个数的指数，如果它们不等，则具有较小指数的操作数的尾数右移以保证两个指数相等：

$$5.6756 \times 10^2 = 0.056\,756 \times 10^4$$

再进行尾数相加：

$$M = 9.985 + 0.056\,756 = 10.041\,756;\ E = 10^4$$

所以和为 $10.041\,756 \times 10^4$。再将和正则化，移动尾数中的十进制小数点，调整指数（如果需要）得到最终的结果：

$$1.004\,175\,6 \times 10^5$$

(2) 尾数相乘而指数相加：

$$M = 2.75 \times 4.5 = 12.375;\ E = -16 + 10 = -6$$

得到乘积 $A \times B = 12.375 \times 10^{-6}$。再将乘积正则化，这里我们假定一个完全正则化的浮点数是小于 10 的，通过调整指数和移动尾数中十进制小数点的位置得到

$$A \times B = 1.2375 \times 10^{-5}$$

在浮点运算中，加法和乘法都会带来舍入误差，而在定点运算中只有乘法才会带来舍入误差。然而，与定点运算不同，在浮点加法中，由于存在很宽的动态范围，很少会发生溢出现象：指数项所包含的比特数越多动态范围就越宽。浮点 DSP 处理器和例程现在已得到广泛的应用（参见第 12 章）。

13.3　ADC 量化噪声和信号质量

ADC 过程将输入信号量化成一个有限位的数，典型值为 8、12 或 16，由此带来了量化噪声。如第 2 章中所讨论的，量化噪声功率（或方差 σ^2_{ADC}）由下式给出：

$$\sigma^2_{\text{A}} = \frac{q^2}{12} = \frac{2^{-2B}}{3} \tag{13.6}$$

其中 q 是量化阶梯的大小，B 是 ADC 的位数。显然，通过增加 ADC 的位数可以确切地降低噪声基底。也可以使用多抽样率技术（参见第 9 章）来降低它。一般来讲，对于 B 超过 12 位的值，量化误差引起的噪声是很小的，除非在类似专业音响等应用场合，这时至少需要 16 位才能达到可接受的性能。

ADC 量化噪声进入 DSP 系统是不可改变的。由于 ADC 而导致 DSP 系统的输出噪声功率由下式给出：

$$\sigma_{oA}^2 = \sigma_A^2\left[\frac{1}{2\pi j}\oint_c H(z)H(z^{-1})\frac{dz}{z}\right] \tag{13.7a}$$

$$= \sigma_A^2 \sum_{k=0}^{\infty} h^2(k) \tag{13.7b}$$

其中

σ_{oA}^2 = 系统输出的ADC量化噪声
$\oint_c$ = 围线积分
$h(k)$ = 系统的冲激响应

方括号中的项目可以看做是"系统功率增益"，它放大（或改变）了ADC噪声，取决于DSP的系统特性。

例13.7 一个8位ADC进入一个DSP系统，其传递函数为

$$H(z) = \frac{1}{z+0.5}$$

估计系统输出的稳态量化噪声功率。

解：
在系统输入处，由于ADC产生的噪声功率为（参见13.6式）

$$\sigma_A^2 = \frac{2^{-16}}{3}$$

输出的ADC噪声功率则为

$$\sigma_{oA}^2 = \sigma_A^2\left[\frac{1}{2\pi j}\oint_c H(z)H(z^{-1})\frac{dz}{z}\right] = \sigma_A^2 \sum_{k=0}^{\infty} h^2(k) \tag{13.8}$$

其中$\oint_c$表示一个围线积分，$h(k)$是系统的冲激响应（参见第4章）。方括号中的项目可被视为系统功率增益，即放大（或改变）了ADC噪声。

对于本例中简单的传递函数，方括号中的项目可以利用第4章讨论的留数法获得。根据第4章的结论，将单位圆作为围线，容易得到方括号中的项等于4/3。因此，量化噪声功率为

$$\sigma_{oA}^2 = \frac{2^{-16}}{3} \times \frac{4}{3}$$

ADC噪声与信号中的固有噪声一起，构成了输入噪声平台。为了保证信号的质量，后续的数字信号处理所产生的失真电平应小于这一噪声基底。在图13.2中，我们给出了一个DSP系统的例子，其中由DSP产生的噪声（本例为IIR滤波）一度超出了输入噪声基底（Wilson, 1993）。在这个例子里，应采用某种补救措施来降低噪声到噪声基底以下（例如使用舍入误差反馈结构或一个足够字长的处理器）。

在实践中，用以表示输入数据的位数是动态范围和信号质量的一种标志。例如，在消费CD音响系统中，音频输入采用16位抽样，对应着一个96 dB的理论输入信噪比（SNR）。为了保持CD的音频质量，DSP的操作应至少具有24位的字长。一个16位字长的处理器能够得到96 dB的动态范围，但考虑到计算误差，实际有效的动态范围会小于这一数字。这样，在对一个特定系统进行有限字长误差分析时，考虑的重要因素是输入信号的质量。

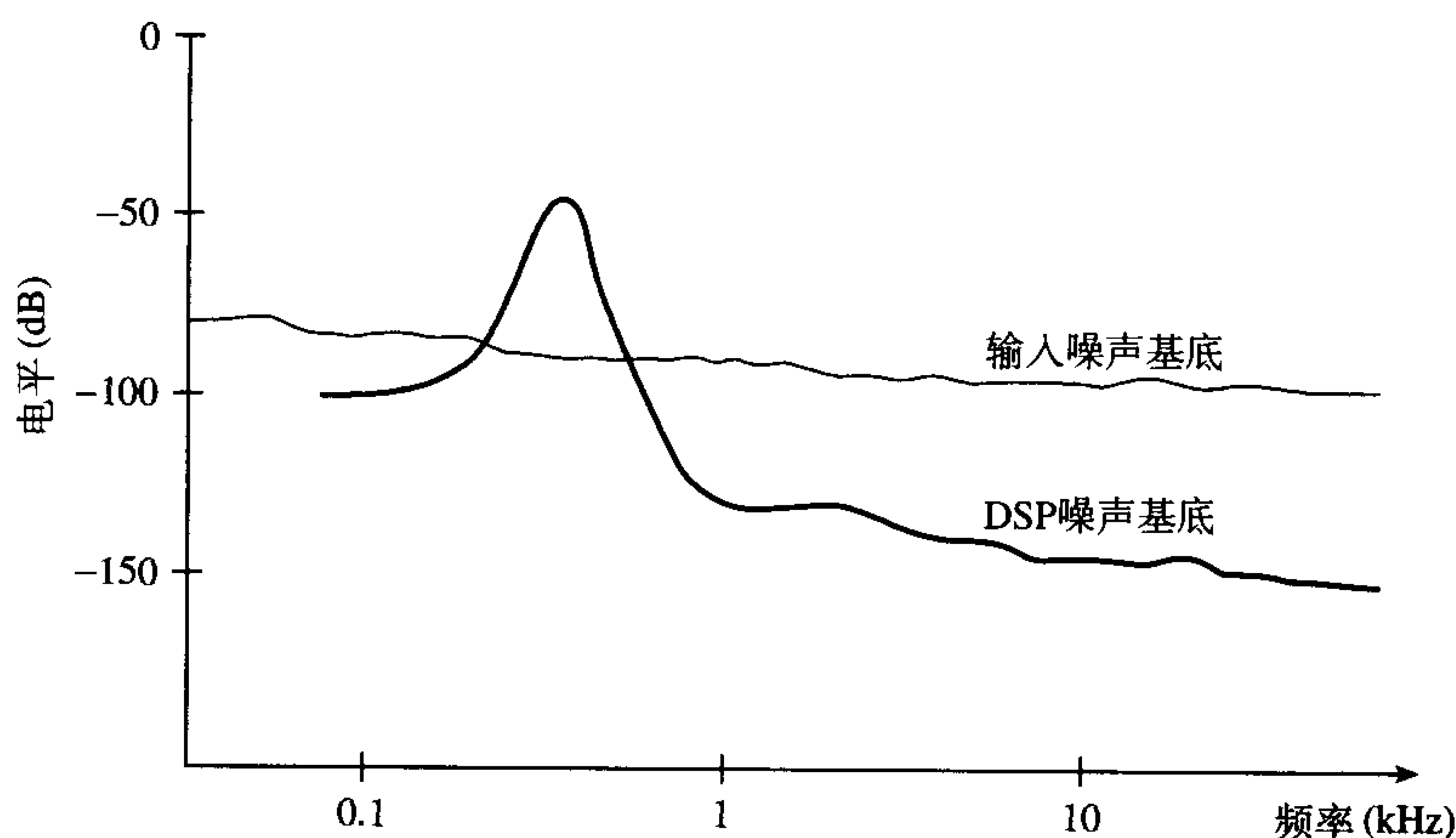


图 13.2　在信号处理中舍入噪声对系统噪声平台的影响示意图（Wilson, 1993）

13.4　IIR 数字滤波器中的有限字长效应

在 2 级 IIR 滤波器设计中，系数 a_k 和 b_k 通常是无限长或具有很高的精度，典型值是 6 个十进制位置。当在一个小系统中实现 IIR 数字滤波器时，例如一个 8 位的微处理器，在滤波器系数表示和执行差分方程所指示的算术操作时会产生误差。这些误差降低了滤波器性能，在某些极端条件下导致不稳定。

在执行 IIR 滤波器之前，非常重要的一点是评估有限字长效应会导致性能下降的程度，并在该下降不可接受时找到解决方法。通常使用更多的位会使这些误差的影响降低到合适的电平，其代价是增加成本。

数字 IIR 滤波器的主要误差有

- ADC 量化噪声，它是由于采用有限位来表示输入数据 $x(n)$ 抽样而产生的；
- 系数量化误差，它是由于采用有限位来表示 IIR 滤波器系数而产生的；
- 溢出误差，它是由于在有限长度寄存器中对部分结果进行加法或累加而产生的；
- 乘法舍入误差，它是由于内部运算结果被舍入（或截断）到容许的字长，从而使输出 $y(n)$ 产生的误差。

滤波器性能下降的程度取决于(i)字长和实现滤波操作所用的算术类型，(ii) 量化滤波器系数和变量所用的方法，以及(iii)滤波器的结构。根据对这些因素的掌握，设计者可以评估有限字长效应对滤波器性能产生的影响，并在需要的时候采取措施。另外有些影响可能是不显著的，这取决于滤波器是如何实现的。例如，在大多数大型计算机上实现一个高级语言程序，系数的量化和舍入误差是不重要的。对于实时处理，有限字长（典型值是 8 位、12 位、和 16 位）用于表示输入和输出信号、滤波器系数和算术操作结果。在这些情况下，基本上都需要分析量化对滤波器性能的影响。

对于 IIR 滤波器，由于它的反馈回路，分析有限字长对性能的影响要比 FIR 滤波器难得多。然而，使用配套的指导手册的 CD 中基于 PC 的程序，可以使我们获得对特定滤波器的实际解决方案。上面列出的四个错误源的影响将在下面几节依次讨论。

13.4.1　滤波器结构对有限字长效应的影响

读者可能早就注意到，IIR 滤波器可以用许多不同的结构加以表示，理论上是完全等价的。但是，在用定点或浮点 DSP 处理器实现它们时，滤波器的表现可能大不相同。

在实际中，IIR 滤波器经常采用二阶直接 I 型和直接 II 型（或标准型，canonic）结构（参见图 13.1 和图 13.2）。直接 I 型具有如下特征：

$$y(n) = \sum_{i=0}^{2} b_i x(n-1) - \sum_{i=1}^{2} a_i y(n-i) \quad \text{– 差分方程}$$

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad \text{– 传递函数}$$

- 5 个滤波器系数
- 4 个延迟单元
- 1 个加法器（4 次加法）
- 对乘积的和进行一次量化
- 一个乘法器（5 次乘法）
- 需要 9 个内存单元以存储数据和系数

标准型结构具有如下特征：

$$\left.\begin{aligned} y(n) &= \sum_{i=0}^{2} b_i w(n-i) \\ w(n) &= x(n) - \sum_{i=1}^{2} a_i w(n-i) \end{aligned}\right\} \text{– 两步，差分方程}$$

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad \text{– 传递函数}$$

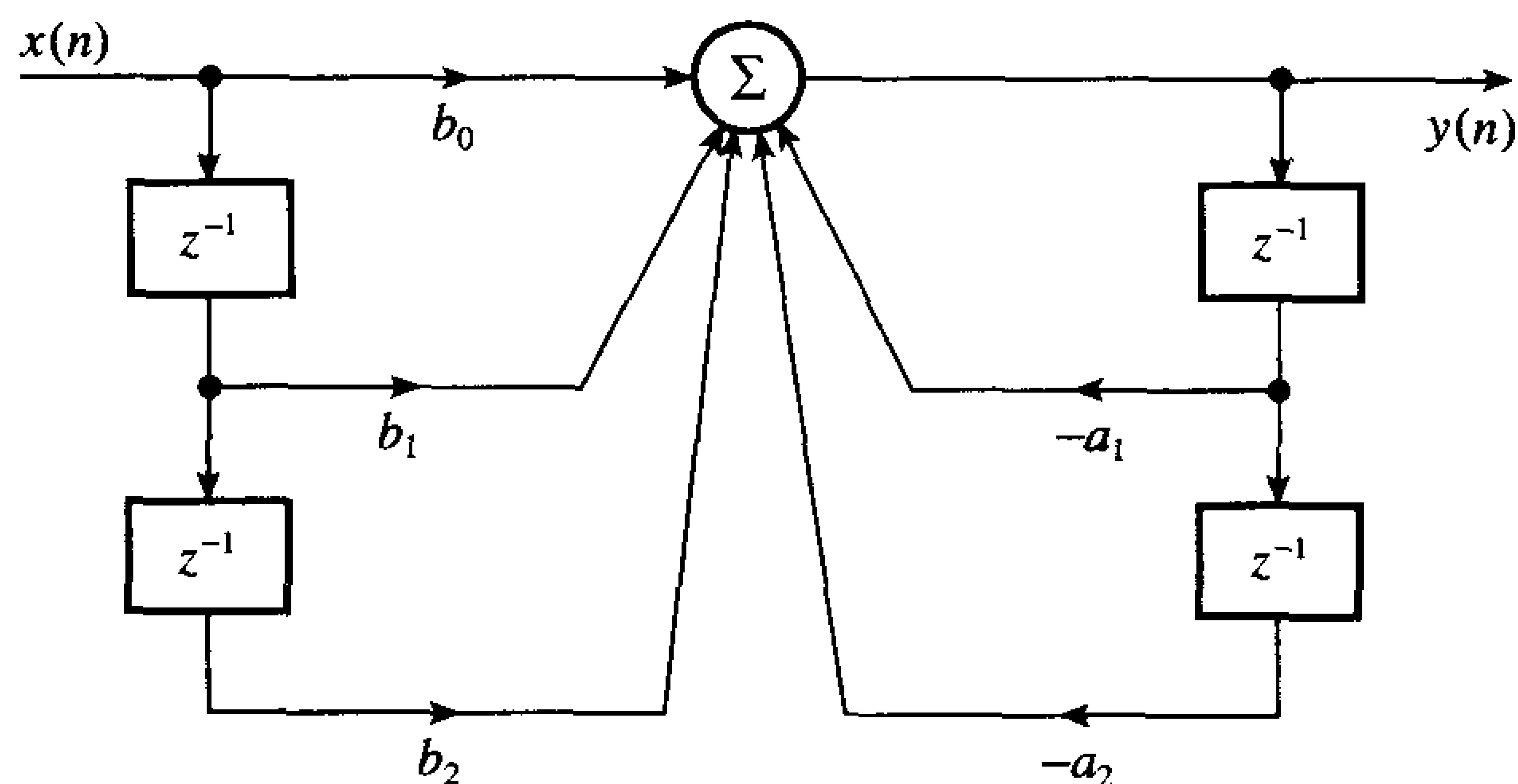
- 5 个滤波器系数
- 2 个延迟单元
- 2 个加法器（4 次加法）
- 对乘积的和进行两次量化
- 一个乘法器（5 次乘法）
- 需要 7 个内存单元以存储数据和系数

我们注意到尽管两个传递函数在理论上是相同的，但是它们之间存在重要的区别。例如，在直接 I 型结构（参见图 13.3(a)）中，前馈项（与零点有关）在反馈项（与极点有关）之前。而在标准型结构（参见图 13.3(b)）中刚好相反。在实际中这隐含着标准型结构的极点会放大计算中所产生的噪声。

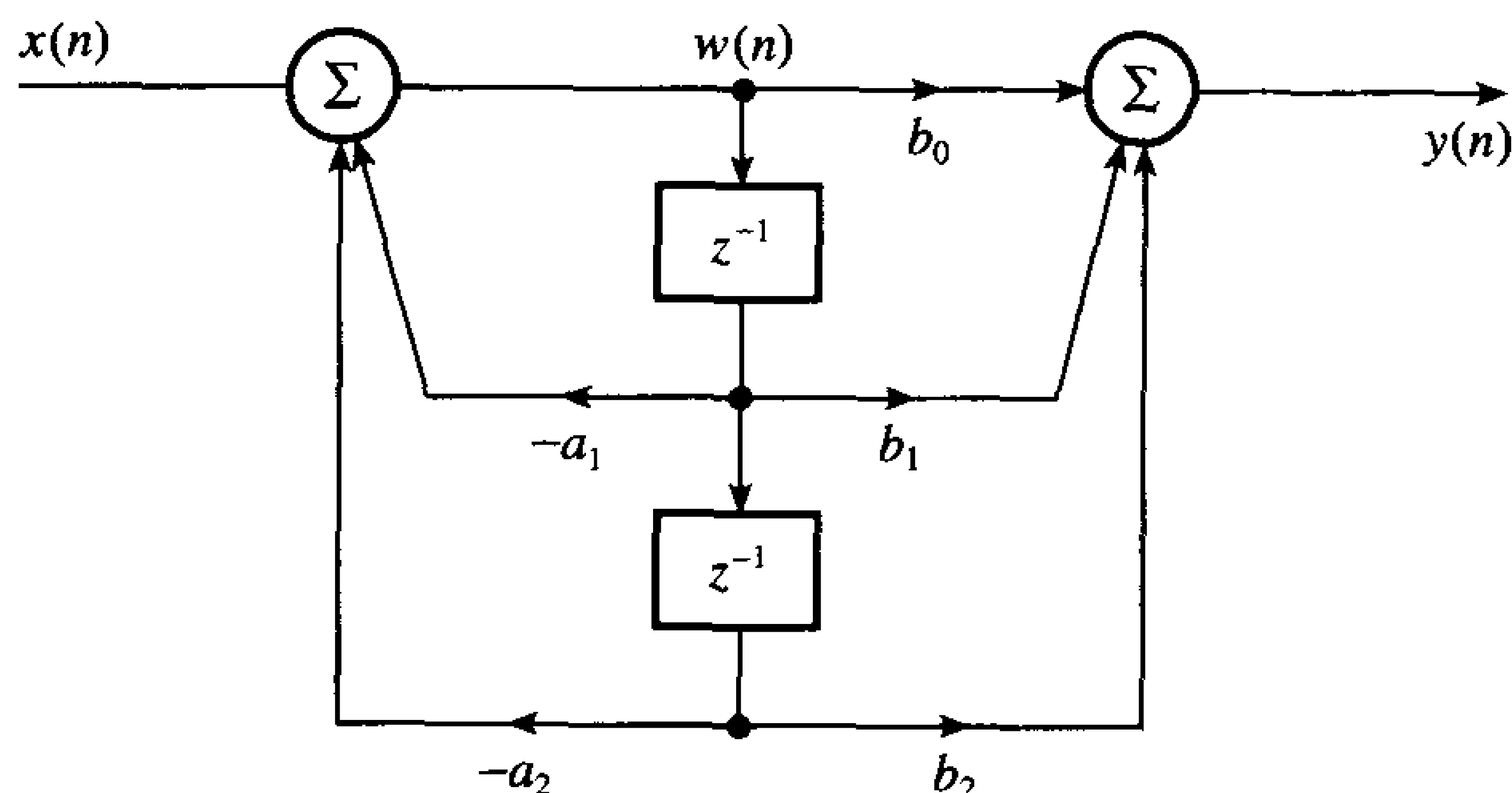
如果直接比较图 13.3(a)和图 13.3(b)，两种结构的差别更为明显。图 13.3(a)的直接 I 型只有一个加法器和一个对乘积和的量化点；而对于标准型结构，图 13.3(b)有两个加法器和 2 个量化点。图 13.3(b)左侧加法器的输出是一个内部节点 $w(n)$。因此标准型可能会产生内部的自激溢出（self-sustaining overflow）。直接 I 型没有内部节点，进行补码的加法算术产生的输出溢出可以通过自我修正或比较容易的方法解决。另外，其输入 $x(n)$的幅度由系数 b_0 确定，而不像标准型结构的输入是无限制的。所以，我们看到用以实现 DSP 功能（本例为 IIR 滤波器）的结构对 DSP 系统的最终性能具有非常重要的影响。

在实际应用中，高阶 IIR 滤波器是采用二阶子滤波器的串联或并联组合来实现的，请参见图 13.4。在串联实现的连接中出现了三个困难：

- 怎样让分子和分母配对以确定各二阶子滤波器的系数；
- 各二阶子滤波器之间连接的排序；
- 在复合滤波器中，需要限制各节点处的信号幅度，以确保在容许的字长范围内。



(a) 直接I型二阶滤波器



(b) 标准型二阶滤波器

图 13.3　IIR 滤波器的基本结构框图

如第8章所讨论的，对滤波器各部分的配对和排序与有限字长效应联系在一起。根据滤波器的阶数，对各二阶子滤波器进行不同的配对和排序得到很多可能的等效滤波器，但它们受到的有限字长效应并不相同。配套的指导手册中的有限字长分析程序（Ifeachor, 2001）可以用来确定适当的滤波器配置。

对于并联实现，各子系统连接的顺序并不重要。

13.4.2　IIR 数字滤波器中的系数量化误差

IIR 滤波器特征由下面的方程给出：

$$H(z)=\frac{\sum_{k=0}^{N}b_kz^{-k}}{1+\sum_{k=1}^{M}a_kz^{-k}}$$

$$[H(z)]_q=\frac{\sum_{k=0}^{N}[b_k]_qz^{-k}}{1+\sum_{k=1}^{M}[a_k]_qz^{-k}}$$

其中

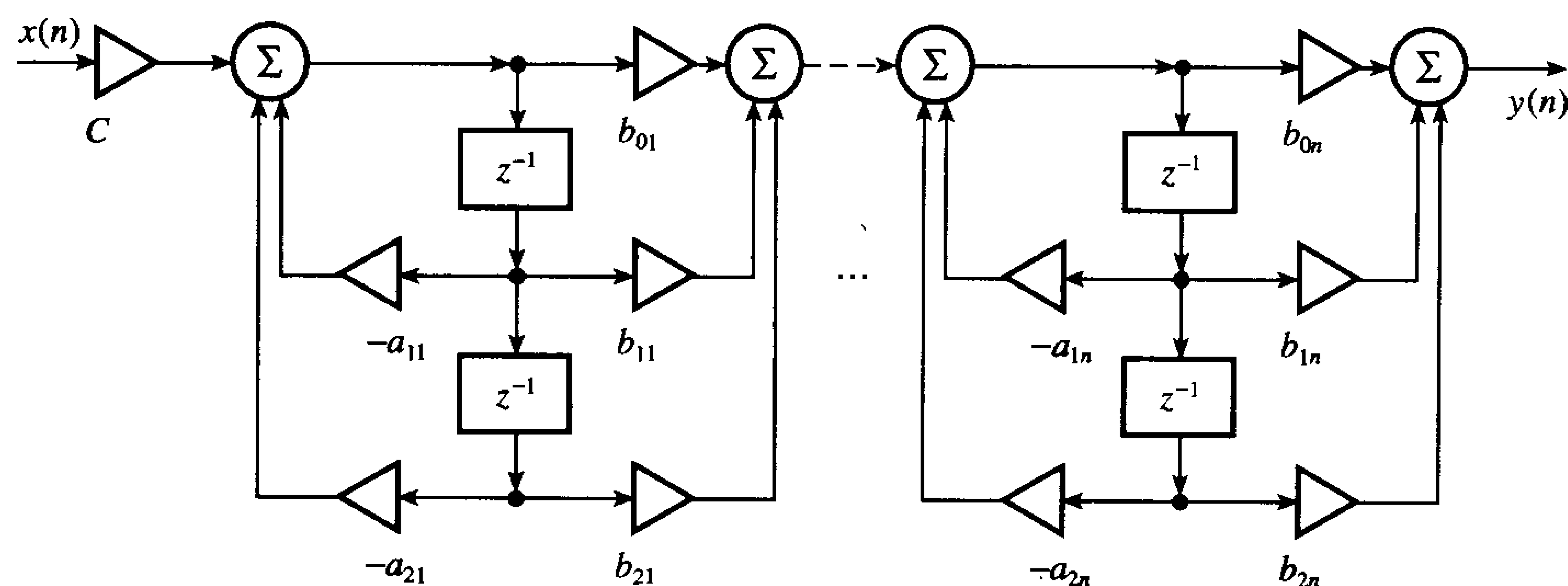
$$[a_k]_q = a_k + \Delta a_k; \quad [b_k]_q = b_k + \Delta b_k$$

Δa_k, Δb_k 是系数 a_k 和 b_k 各自的变化

q 表示一个量化的数量

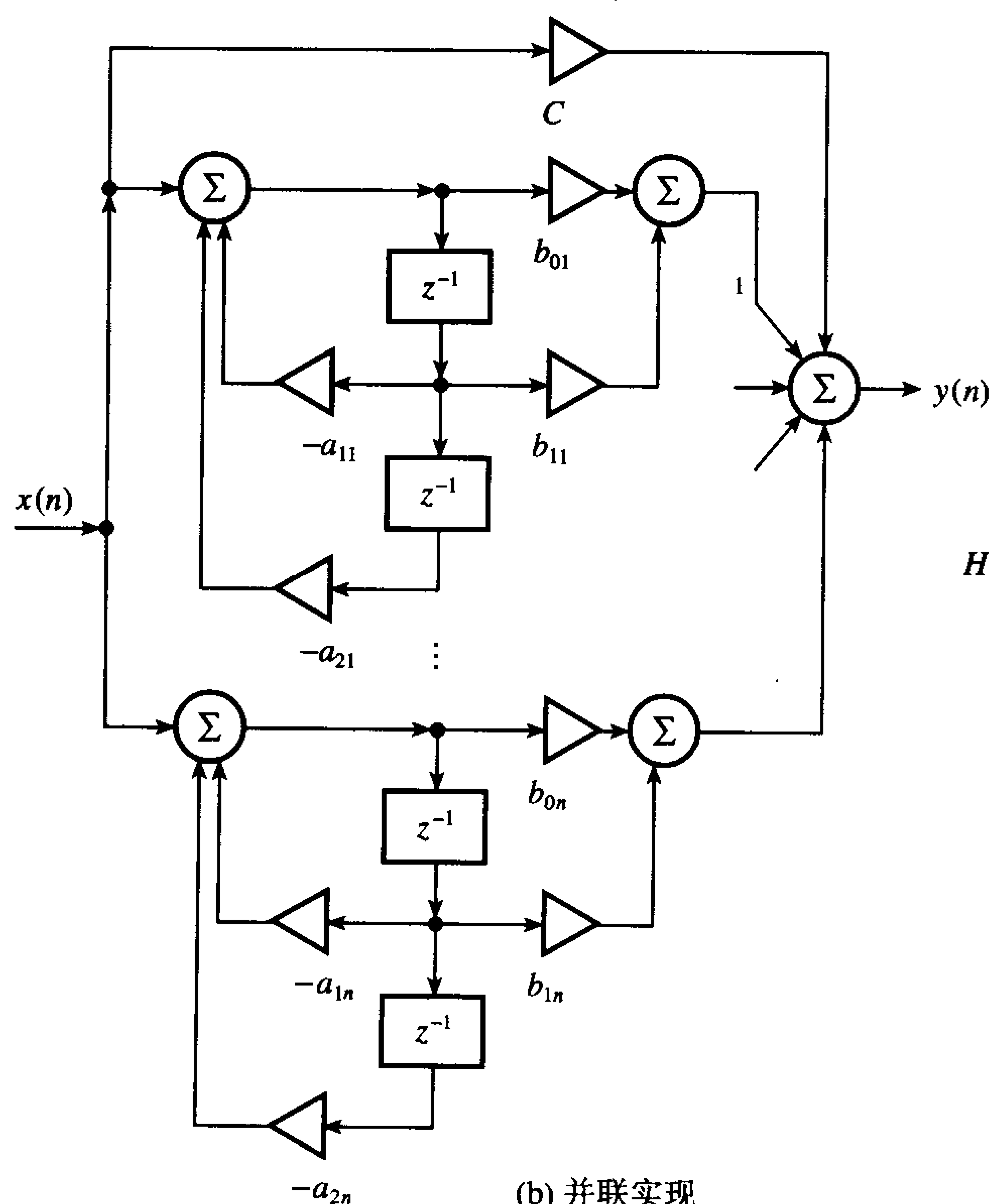
将滤波器系数量化成一个有限位数的主要结果是改变了 z 平面上 $H(z)$ 的极点或零点位置。例如对于窄带滤波器，极点非常靠近单位圆，因此任何对它们位置的改变都可能导致滤波器不稳定。用以表示系数的位数越少，极点和零点位置的改变会越大。

除了潜在的不稳定，极点和零点位置的变化还会导致频率响应（或称频响）的变化。应该分析量化后的滤波器特性，以确保它的字长能得到稳定和满意的频率响应。因此，设计者在本阶段应确定用以表示滤波器系数的位数，以从根本上保证稳定和希望的频率响应。



$$H(z) = C \times \prod_{k=1}^{n} \frac{b_{0k} + b_{1k}z^{-1} + b_{2k}z^{-2}}{1 + a_{11}z^{-1} + a_{21}z^{-2}}$$

(a) 串联实现



$$H(z) = C + \sum_{k=}^{n} \frac{b_{0k} + b_{1k}z^{-1}}{1 + a_{1k}z^{-1} + a_{2k}z^{-2}}$$

(b) 并联实现

图 13.4　高阶 IIR 滤波器的实现结构

例 13.8　用一个带通数字 IIR 滤波器为一个 4.8 kb/s 的调制解调器进行数字时钟恢复，其传递函数由下式确定：

$$H(z)=\frac{1}{1+a_1z^{-1}+a_2z^{-2}}$$

其中

$$a_1=-1.957\,558,\quad a_2=0.995\,813$$

假定抽样频率为 153.6 kHz，评估将系数量化成 8 位对极点位置和中心频率产生的影响。

解：

首先我们找到没有量化时滤波器的极点位置。极点位置是（参见 13.10 式）

$$r=\sqrt{0.995\,913}=0.997\,95,\quad \theta=\cos^{-1}\left(\frac{1.957\,558}{2r}\right)=11.25°$$

它对应的中心频率为 4.7999 kHz（$153.6\times10^3\times11.25/360$）。

然后我们将系数量化到 8 位。由于其中一个系数大于 1，我们指定 1 位为符号位，1 位给系数的整数，6 位给系数的小数部分。系数量化得到

$$a_1'=-1.957\,558\times2^6=-125\equiv10000011$$

$$a_2'=0.995\,913\times2^6=63\equiv00111111$$

用小数表示，量化系数的值为

$$a_1'=-\frac{125}{64}=-1.953\,125;\quad a_2'=\frac{63}{64}=0.984\,375$$

新的极点变成

$$r'=0.992\,156;\quad \theta'=10.171\,853°$$

而中心频率则变成

$$f_0=\left(\frac{10.171\,853}{360}\right)\times153.6\times10^3=4.3399\text{ kHz}$$

13.4.3　稳定和满足频响所需要的系数字长

我们对稳定的讨论限制于二阶子滤波器，因为它是任意滤波器的基本组成部分。考虑一个如下方程所确定的二阶滤波器：

$$H(z)=\frac{b_0+b_1z^{-1}+b_2z^{-2}}{1+a_1z^{-1}+a_2z^{-2}}$$

$$y(n)=\sum_{k=0}^{2}b_kx(n-k)-\sum_{k=1}^{2}a_ky(n-k)$$

极点（或分母的根）位于

$$p_1=\tfrac{1}{2}[-a_1+(a_1^2-4a_2)^{1/2}]\tag{13.9a}$$

$$p_2=\tfrac{1}{2}[-a_1-(a_1^2-4a_2)^{1/2}]\tag{13.9b}$$

对每一个二阶子系统，有三种类型的极点：复共轭极点、不相等实极点以及相等（多阶）实极点。复共轭极点是最常见的，其产生条件为 $a_1^2<4a_2$。对于本例，极点的位置在从原点出发、半径为 r、角度为 θ 的圆上，

$$p_1=r\angle\theta,p_2=r\angle-\theta\tag{13.10}$$

其中

$$r = a_2^{1/2},\ \theta = \cos^{-1}\left(-\frac{a_1}{2r}\right)$$

由于系数量化，a_1和a_2发生了很小的变化，但这还是会导致r和θ发生变化。从稳定性出发，滤波器系数必须在三角形稳定区以内（参见图13.5），边界为

$$0 \leqslant |a_2| < 1 \tag{13.11a}$$

$$|a_1| \leqslant 1 + a_2 \tag{13.11b}$$

第一个边界条件限制极点必须在单位圆内，因为极点的半径由13.10式给出。对于13.10式和13.11式，可以推导出许多简单的公式来估计维持稳定性所需的位数，但它们都只能在有限的情况下使用。另一种为稳定性估计合适的滤波器字长的方法是，分别单独分析二阶子系统在不同滤波器字长值下的性能（参见例13.9）。

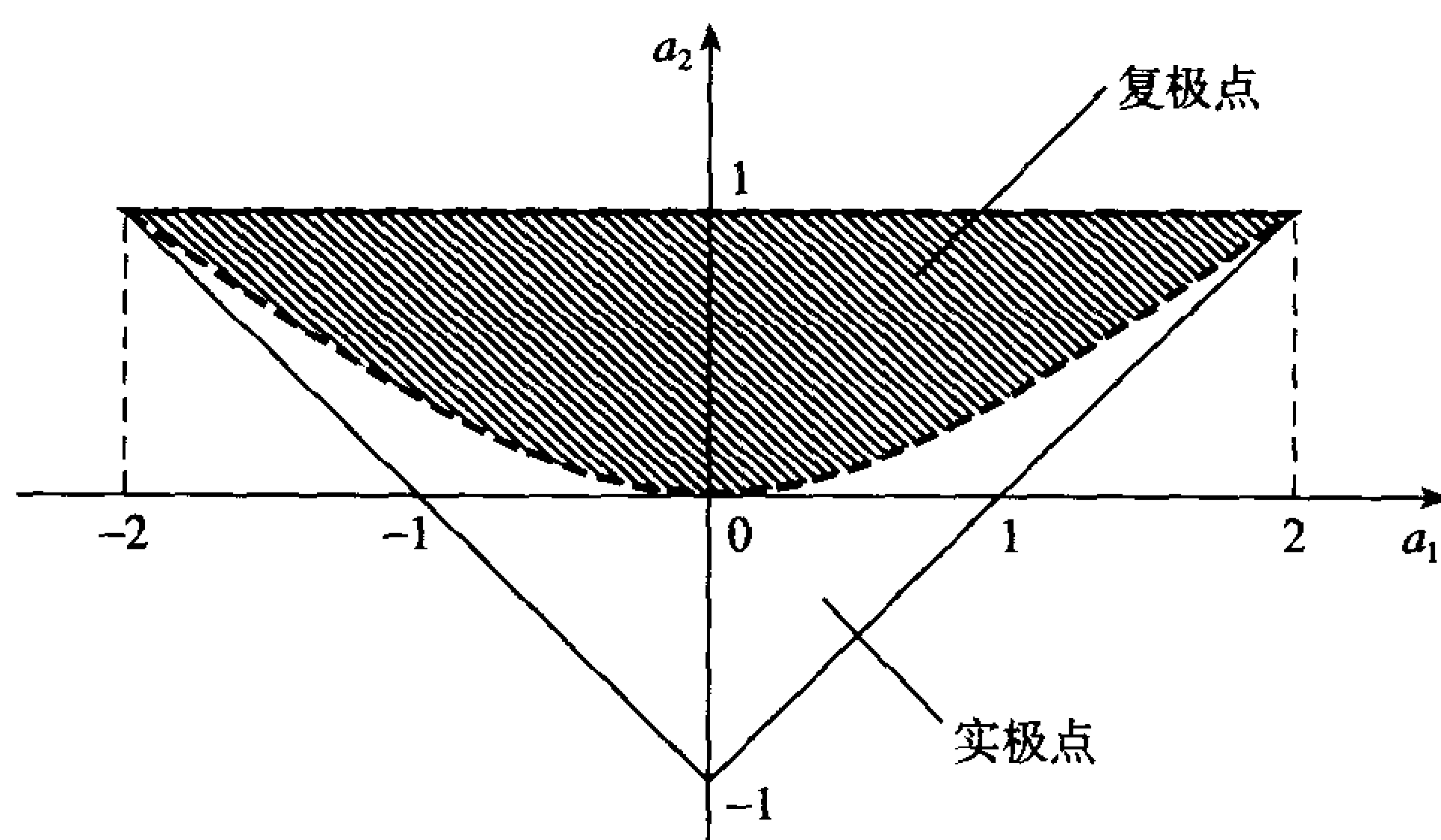


图13.5 三角形稳定区显示了滤波器系数a_1和a_2的值，这时滤波器是稳定的

为稳定所需的位数并不能保证得到满意的响应。使用过小的位数表示系数会改变通带和阻带的响应（参见图13.6）。通带的改变主要来自于极点位置的改变，而阻带的改变则主要来自于零点位置的改变。

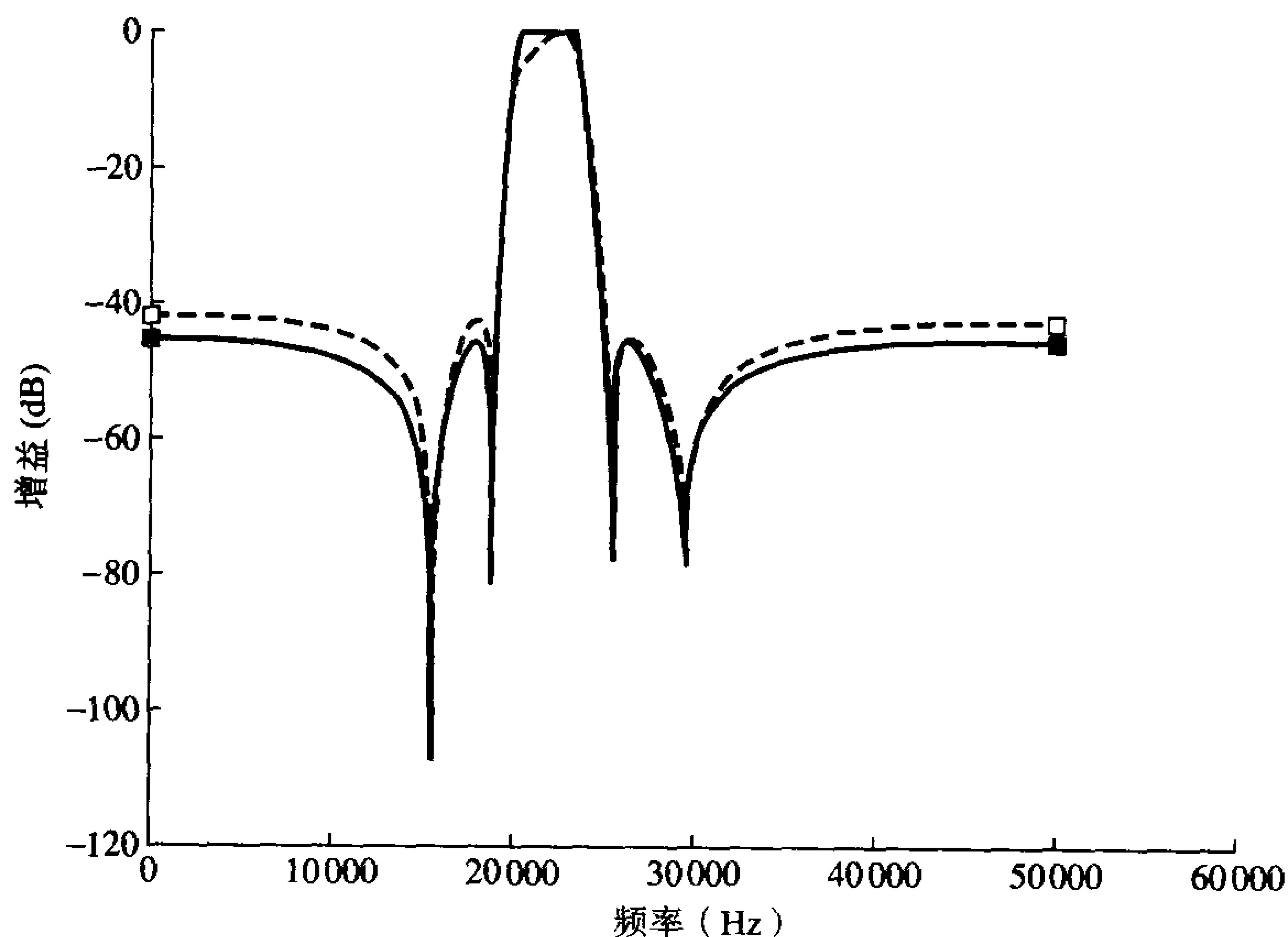


图13.6 系数量化对频率响应产生的实际影响：■，未量化；□，5位量化

为了确定一个合适的系数字长以达到满意的频率响应,应该找到满足通带和阻带需求的最小系数字长。虽然它涉及大量的计算，当前广泛使用PC和有限字长分析（FWA）程序使确定特定滤波器字长变得相对容易（指导手册的CD中有一个示例程序）。另一种方法是利用统计原理，同样也能产生相当精确的估计（Antoniou, 1979）。

如果在量化滤波器系数时使用了优化方法,可能只用较少的位数就可以表示滤波器系数,或者增加滤波器阶数而所有其他系数保持不变。它使滤波器阶数与系数字长之间可以取得一种平衡（Rabiner and Gold, 1975）。例如，在一个特定问题中，设计者可能希望使用一个字长已确定的处理器或系统。如果设计者发现所需的字长超过了处理器字长,那他可以通过增加滤波器阶数使所需系数字长下降以符合处理器的条件。然而，使用高阶滤波器需要更多的计算量，这将影响到速度，以及可能产生更多的舍入噪声。设计者需要仔细考虑其中的平衡问题。

例 13.9　一个数字滤波器需要满足下面的频响要求：

通带	20.5 ~ 23.5 kHz
阻带	0 ~ 19 kHz, 25 ~ 50 kHz
通带波纹	≤ 0.25 dB
阻带衰减	> 45 dB
抽样频率	100 kHz

(1) 确定滤波器合适的传递函数。

(2) 确定一个合适的系数字长以

　(a) 保持稳定性；

　(b) 满足频响需求。

(3) 计算和绘出未量化滤波器的频率响应，以及(2)对应的量化后滤波器的频率响应。

解：

(1) 利用设计程序（指导手册的CD中，详见前言），得到满足设计要求的如下传递方程所确定的椭圆滤波器：

$$H(z) = H_1(z)H_2(z)H_3(z)H_4(z)$$

其中

$$H_1(z) = \frac{1 + 0.0339z^{-1} + z^{-2}}{1 - 0.1743z^{-1} + 0.9662z^{-2}}$$

$$H_2(z) = \frac{1 - 0.7563z^{-1} + z^{-2}}{1 - 0.5588z^{-1} + 0.9675z^{-2}}$$

$$H_3(z) = \frac{1 + 0.5331z^{-1} + z^{-2}}{1 - 0.2711z^{-1} + 0.9028z^{-2}}$$

$$H_4(z) = \frac{1 - 1.1489z^{-1} + z^{-2}}{1 - 0.4441z^{-1} + 0.9045z^{-2}}$$

(2) (a) 每个二级子滤波器的分母系数被舍入量化成 B 位（$B = 2, 3,..., 29$），其中包括符号位。对于 B 的每个值，都要计算量化系数和极坐标下的极点位置。为说明这一点，考虑二阶子滤波器 $H_1(z)$。对于 $B = 8$ 位，将分母系数舍入量化成：

$$a_1 = -(0.1743 \times 2^7 + 0.5) = -22.8104 = -22$$

$$a_2 = 0.9662 \times 2^7 + 0.5 = 124.1736 = 124$$

系数用小数表示为

$$a_1 = -22/128 = -0.171\,875$$

$$a_2 = 124/128 = 0.968\,75$$

根据13.10式，$B=8$ 位部分的极点矢径和相角为

$$r = \sqrt{0.968\,75} = 0.9843,$$

$$\theta = \cos^{-1}\left(-\frac{b_1}{2r}\right) = \cos^{-1}(0.087\,308) = 84.99°$$

所有的量化系数和极坐标都可用分析程序来计算。如果对于任何一个系数字长，子滤波器的极点矢径大于或等于单位1，则有着潜在的不稳定。对于所有子滤波器，至少需要 $B=5$ 位来保持稳定性。一般来说，如果一个未量化二阶子滤波器的极点半径 $r<0.9$，系数字长为8位或更多就不大可能出现不稳定。

(b) 二阶子滤波器的每个系数被量化成上面给出的不同字长。对于每个字长，量化后的系数重新组合成一个完全量化的直接形式的传递函数。字长分别为5位和12位的例子在表13.4中给出。按不同系数字长量化后滤波器的通带波纹和阻带衰减也可得到。可以看到，为了同时满足通带和阻带频率响应的技术指标要求，至少需要16位。我们注意到，这要大于稳定所要求的字长。

表13.4　例13.9的系数和字长效应

	$B(k)$			$A(k)$		
k	理想值	5位	16位	理想值	5位	16位
0	1.000 000	1.000 000	1.000 000	1.000 000	1.000 000	1.000 000
1	−1.338 200	−1.250 000	−1.338 165	−1.448 300	−1.437 500	−1.448 273
2	3.806 737	3.707 031	3.806 700	4.483 108	4.355 469 0	4.483 071
3	−3.556 357	−3.288 574	−3.556 255	−4.220 527	−4.060 791	−4.220 431
4	5.629 177	5.443 726	5.629 105	6.647 162	6.261 536	6.647 087
5	−3.556 357	−3.288 574	−3.556 255	−3.945 450	−3.677 216	−3.945 354
6	3.806 737	3.707 031	3.806 700	3.918 398 1	3.573 486	3.918 352
7	−1.338 200	−1.250 000	−1.338 165	−1.182 602 0	−1.067 047	−1.182 575
8	1.000 000	1.000 000	1.000 000	0.763 340 2	0.672 912	0.763 338

(3) 对未量化和量化（$B=5$ 位）滤波器，经伸缩变换后最大为0 dB的频响特性在图13.6中给出。在视觉上，按16位量化的滤波器响应与未量化滤波器是相同的，因此没有给出。

13.4.4　加法溢出误差及其影响

在2的补码运算中，两个相同符号的大数相加可能产生溢出，因为和可能超出了容许的字长，结果导致输出抽样符号的改变。由此一个很大的正数变成了一个很大的负数，反之亦然（参见图13.7）。考虑图13.8的标准型子滤波器。由于IIR滤波器的递归本质，在 $w(n)$ 处的溢出被反馈和用于计算下一个输出，从而导致更多的溢出，产生了意料之外的自激振荡。它们又被称为大规模溢出极限环（large-scale overflow limit cycle），一旦出现就很难停止，除非重启动滤波器。

极限溢出产生于加法器的输出端，可以通过限制加法器的输入大小来防止，其代价则是降低了信噪比（SNR）。因此重要的是选择合适的伸缩因子，在防止溢出的同时在最大程度上保持其SNR。

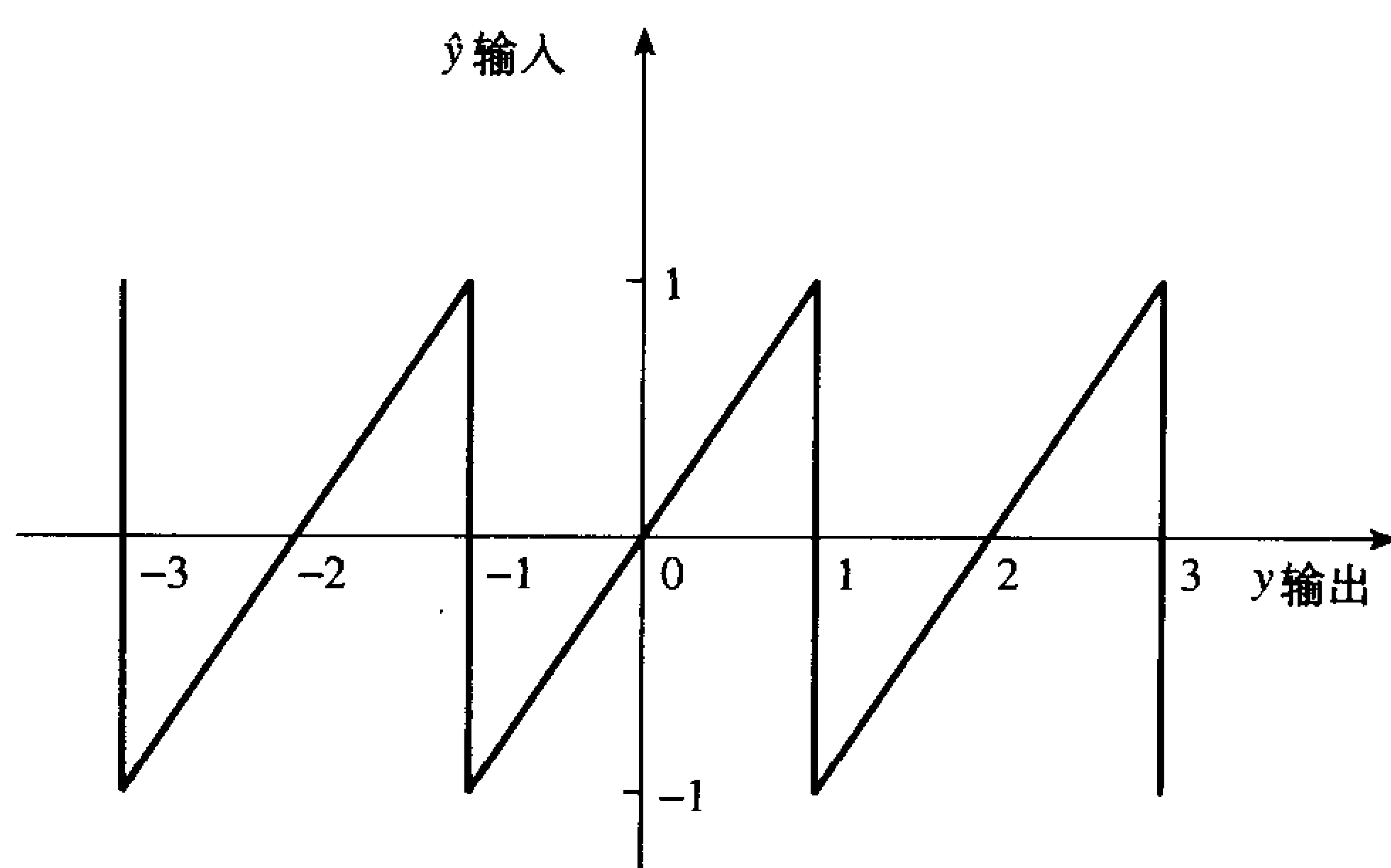


图 13.7　2 的补码运算的溢出特性。当输入超出了容许的范围（−1, 1）时产生了溢出

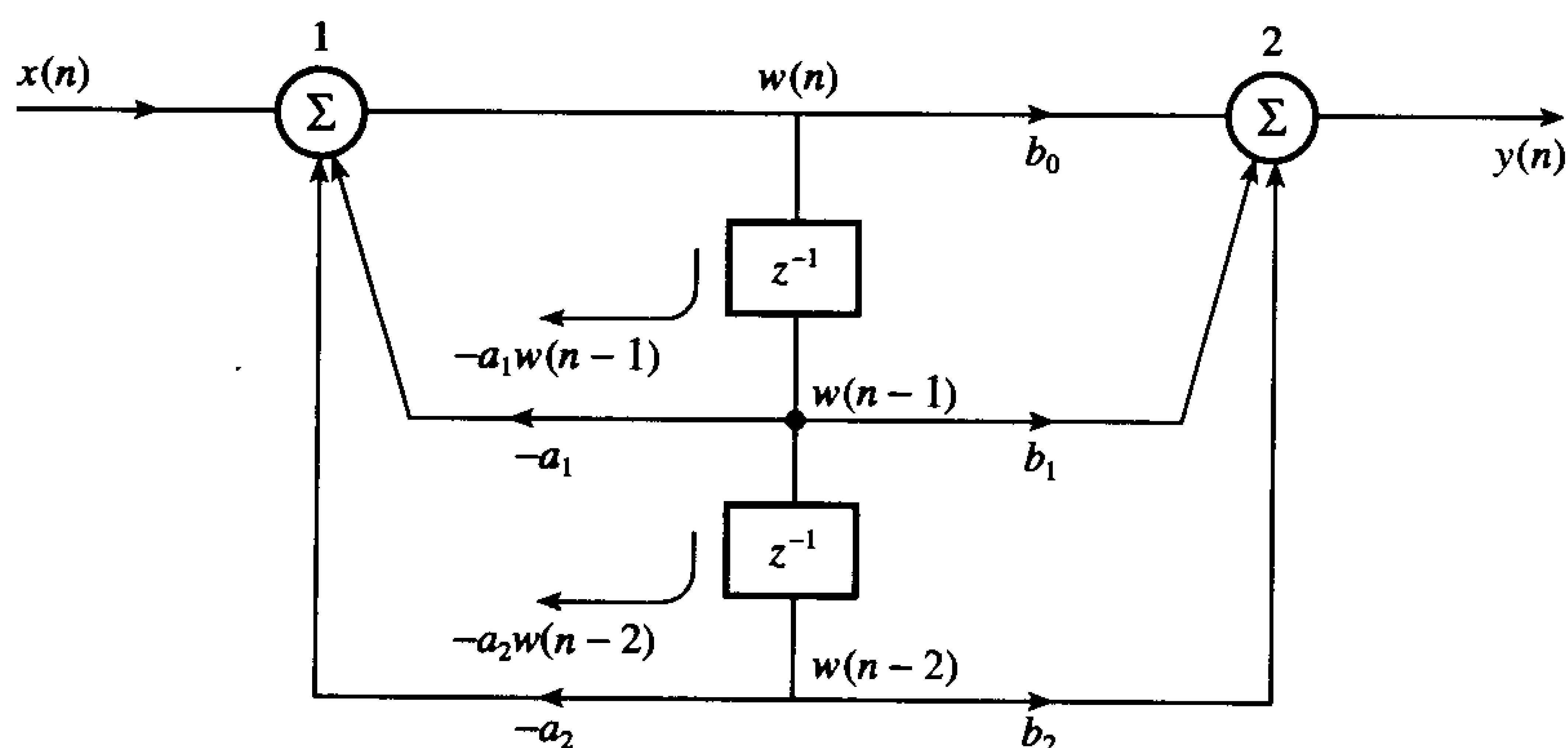


图 13.8　加法溢出效应的示意图。加法器1处输入大的同符号数会导致$w(n)$过大。由于$w(n)$存在反馈，该效应是自激的

13.4.5　伸缩变换原则

13.4.5.1　标准子滤波器

考虑图13.9(a)的二阶标准子滤波器。在滤波器输入端的伸缩因子为s_1，应选择其值避免或降低左端加法器输出溢出的可能性。为了保持总的滤波器增益不变，将分子系数乘以s_1。

有三种确定一个滤波器的合适伸缩因子的通用方法。方法 1，通常称为L_1范数型（L_1 norm），伸缩因子由下式给出：

$$s_1 = \sum_{k=0}^{\infty} |f(k)| \tag{13.12}$$

其中$f(k)$是从输入到第一个加法器输出的冲激响应，即$w(n)$。伸缩因子s_1保证从滤波器输入到$w(n)$的总增益是1，因此在$w(n)$处不可能发生溢出。冲激响应$f(k)$可以通过确定响应的传递函数$F(z)$，再做z的反变换而获得。

方法 2，通常称为L_2范数型（L_2 norm），伸缩因子s_1为

$$s_1 = \left[\sum_{k=0}^{\infty} f^2(k)\right]^{1/2} \tag{13.13}$$

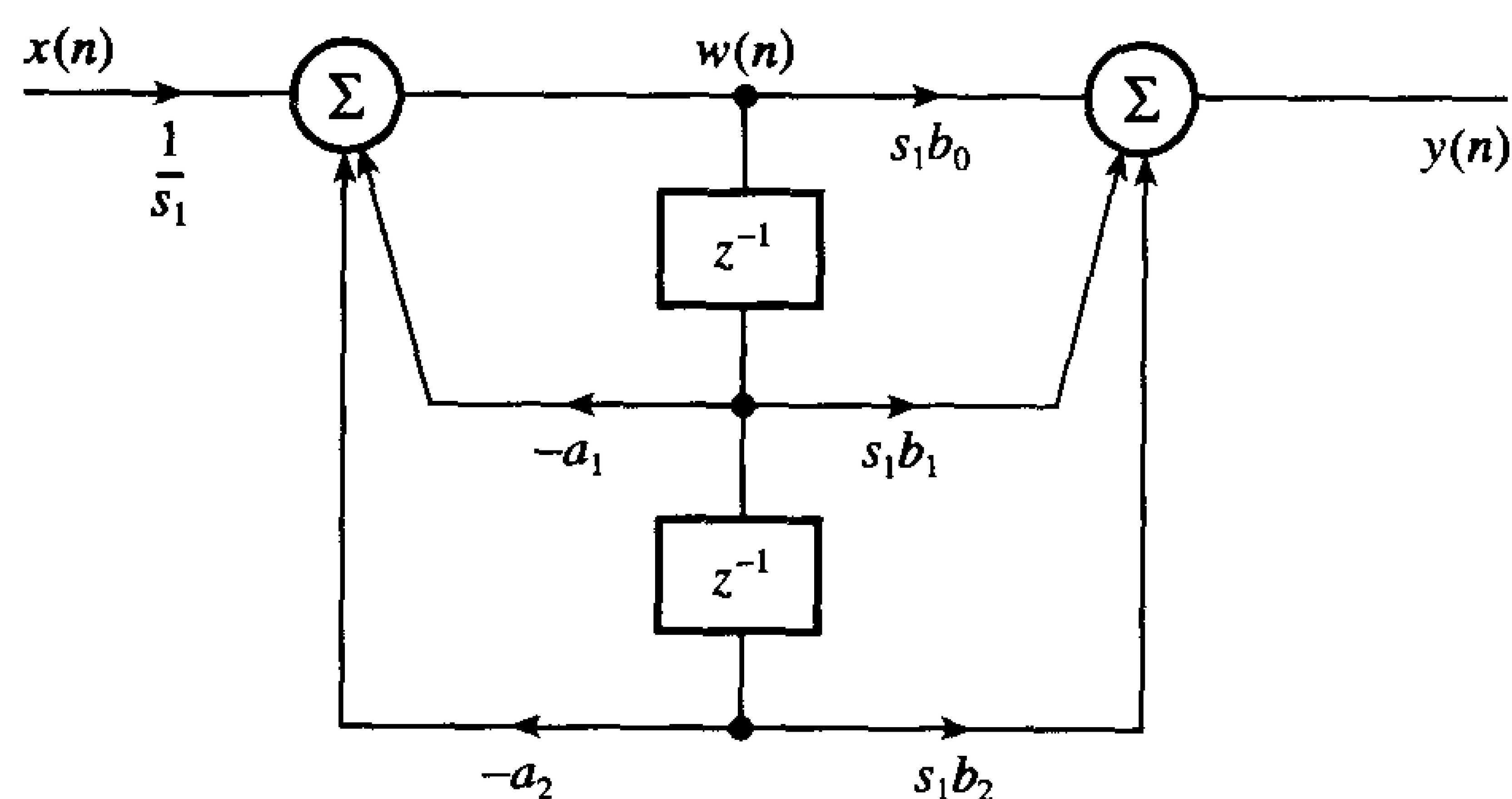
另一个获得L_2范数型伸缩因子的方法是通过以下的围线积分：

$$\sum_{k=0}^{\infty} f^2(k) = \frac{1}{2\pi j}\oint F(z)F(z^{-1})\frac{dz}{z} \tag{13.14}$$

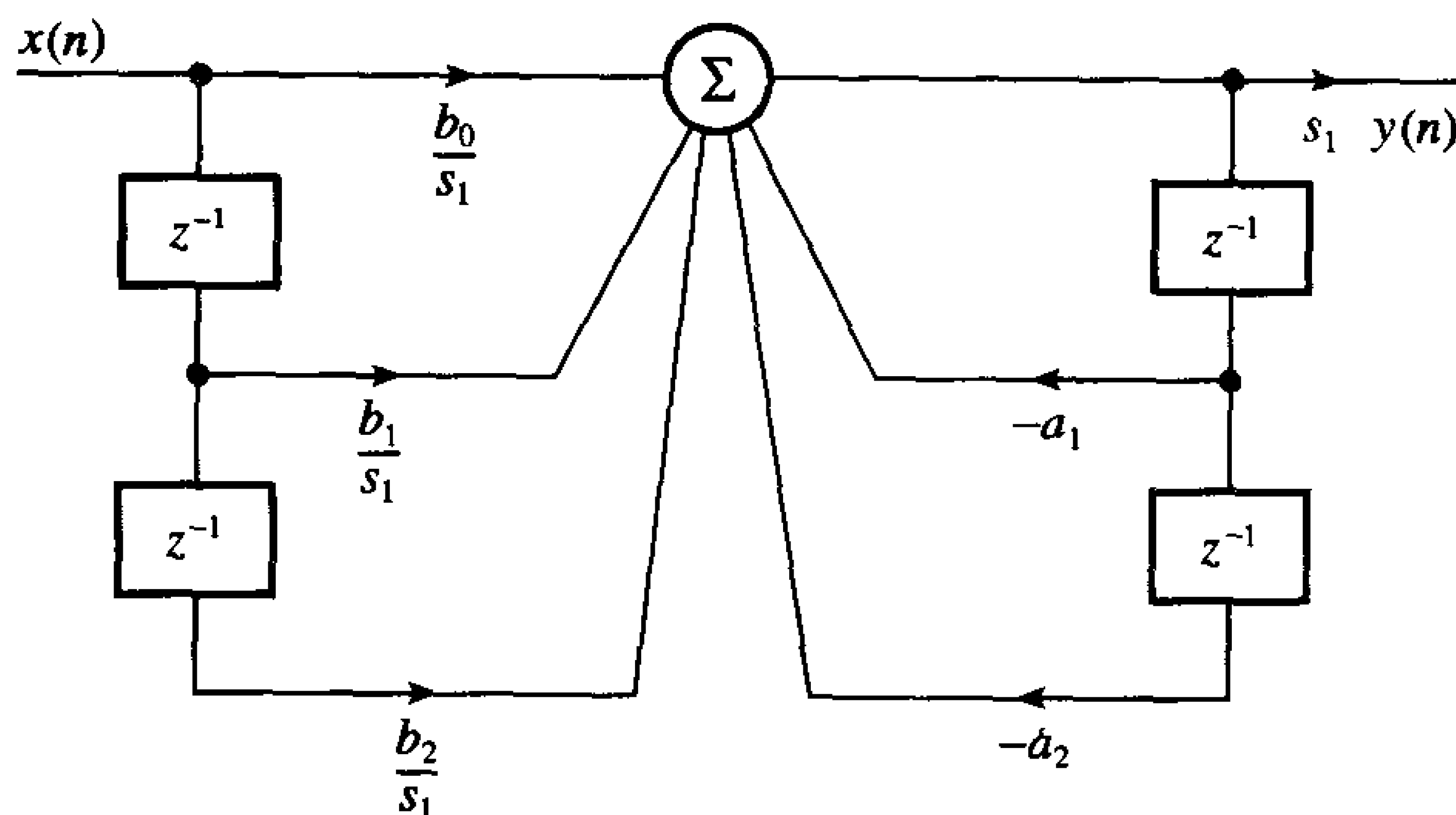
其中$F(z)$是$f(k)$的z变换，$\oint$表示一个沿着单位圆$|z|=1$的围线积分。求其值，我们得到（参见附录13B）

$$\begin{aligned} s_1^2 = \sum_{k=0}^{\infty} f^2(k) &= \frac{1}{2\pi j}\oint \frac{1}{1+a_1z^{-1}+a_2z^{-2}}\frac{1}{1+a_1z+a_2z^2}\frac{dz}{z} \\ &= \frac{1}{1-a_2^2-a_1^2(1-a_2)/(1+a_2)} \end{aligned} \tag{13.15}$$

使用13.15式可以避免计算13.13式的无限次求和。然而在实际中，$f(k)$只有一定量个数的有效值，可以通过一个适当的有限字长分析程序来计算。



(a) 标准型



(b) 直接型

图13.9　二阶子滤波器的伸缩变换原则

方法3，即L_∞范数型（L_∞ norm），其伸缩因子为

$$s_1 = \max|F(w)| \tag{13.16}$$

其中$F(w)$是输入与$w(n)$之间频率响应的峰值幅度。

方法1的潜在假设是输入范围为有限的，即$|x(n)|<1$。这种伸缩变换措施可以确保无论哪种输入类型都不会产生溢出。某种程度上它是一个过于激烈的伸缩变换措施，因为它适用的场合不大可能存在于正常的现实世界。L_2范数型对应着同时对输入和传递函数加上一个能量约束。它主要吸引人的地方是有限字长效应分析需要计算L_2范数型（比较例13.8和13.14式），而且对于大多数滤波器结构它都能推导出闭式表达。方法3确保当输入为一个正弦波时，滤波器不会产生溢出，从而成

为一个折中选择。它是最经常选用的伸缩变换措施，尤其是它允许我们在试验中利用正弦波来检测其伸缩变换性能。

一种第 i 个伸缩因子的简洁表达式为

$$s_i = \|F\|_p$$

其中符号 $\|\cdot\|$ 代表求范数，$p = 1, 2, \infty$ 代表范数的类型。用 3 种方法获得的伸缩因子满足下面的关系：

$$L_2 < L_\infty < L_1$$

13.4.5.2　直接结构

考虑图 13.9(b)给出的直接结构。由于滤波器只有一个加法器，内部溢出不再是一个问题，所以输入的伸缩变换不是严格必要的，这是直接结构一个吸引人的地方。中间溢出可能发生在计算加法器输出 $y(n)$的过程中。在最终输出没有溢出的条件下，它是无关紧要的。如果需要，可以在图 13.9 中使用伸缩变换措施。

例 13.10　确定一个合适的伸缩因子，防止或降低具有下面传递函数的 IIR 低通滤波器发生溢出的可能性：

$$H(z) = \frac{1 + 2z^{-1} + z^{-2}}{1 - 1.058\,135\,9z^{-1} + 0.338\,544z^{-2}}$$

解：

在图 13.10 中给出了滤波器的框图，使用了一个二阶标准结构。利用 FWA 程序（在指导手册的 CD 中——详见前言）来计算 13.12 式、13.13 式和 13.16 式，分别得到三种方法的伸缩因子。结果如下：

	L_1	L_2	L_∞
s_1	3.7112	1.7352	3.5663

作为示例，我们利用 13.15 式计算 L_2 范数型：

$$s_1^2 = \frac{1}{1 - (0.3385)^2 - (1.058)^2[(1 - 0.3385)/(1 + 0.3385)]}$$
$$= 1/0.3322 = 3.01$$
$$s_1 = 1.7350$$

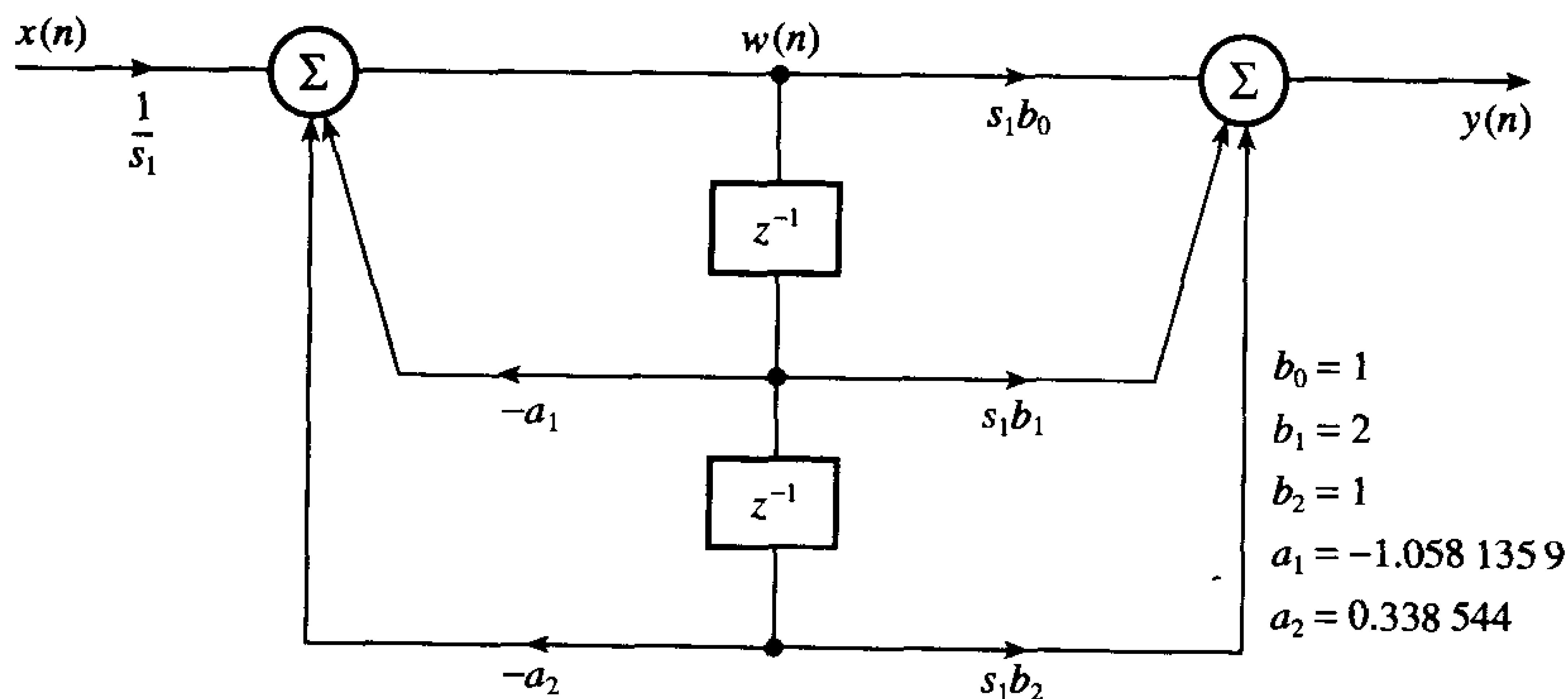


图 13.10　例 13.9 的框图表示

13.4.6 串联实现中的伸缩变换

在实际中，滤波器通过对二阶子滤波器的串联或并联来实现。一个6阶串联实现滤波器的伸缩变换措施如图13.11所示。跟以前一样，选择伸缩因子s_i（$i=1,2,3$），避免或最小化各个子滤波器节点$w_i(n)$处的溢出。对各二阶子滤波器采取的伸缩变换措施与前面单个滤波器的相同。伸缩因子为

$$s_i = \|F_i(z)\|_p \tag{13.17}$$

其中p代表范数的类型：$p=1,2,\infty$。$F_i(z)$是从输入到$w_i(n)$节点的传递函数，由下式给出：

$$F_i(z) = \frac{\prod_{k=1}^{i-1} H_k(z)}{1 + a_{1i}z^{-1} + a_{2i}z^{-2}}, \quad i = 1, 2, 3$$

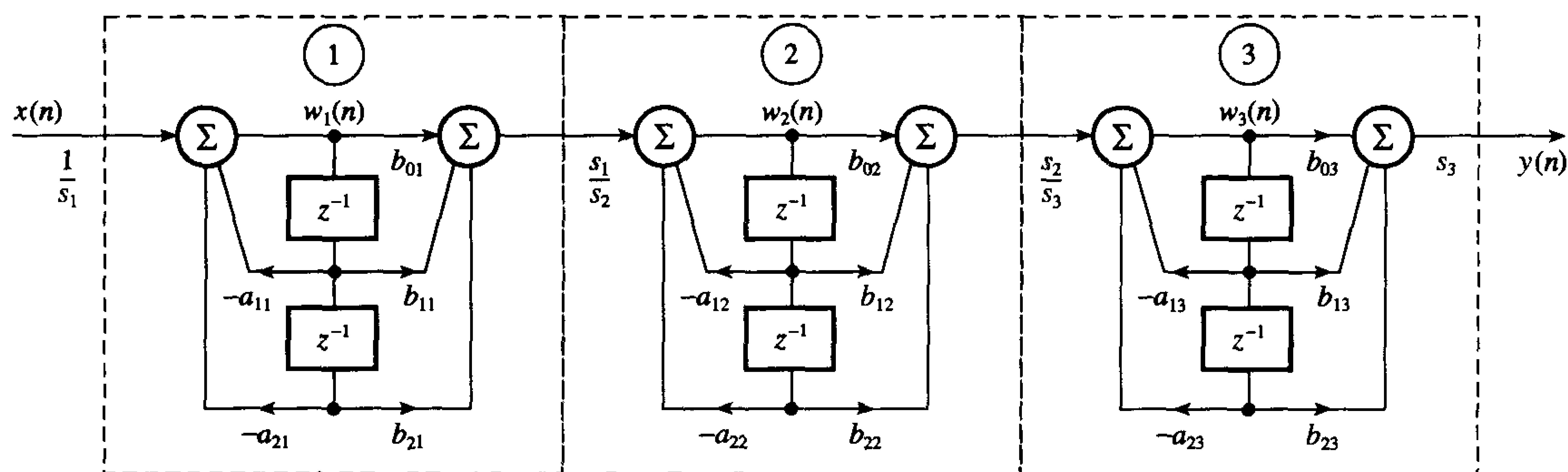


图13.11 对一个6阶IIR滤波器的串联实现进行伸缩变换

对于串联实现，通常的做法是将伸缩因子s_1/s_2放入第一级、将s_2/s_3放入第二级的分子中，依次类推。这样图13.11中的伸缩因子可以重新安排成如图13.12所示的形式。需要注意的是，按前面讨论进行伸缩变换后，滤波器的传递函数必须同未伸缩变换前滤波器的相同（至少在理论上）。

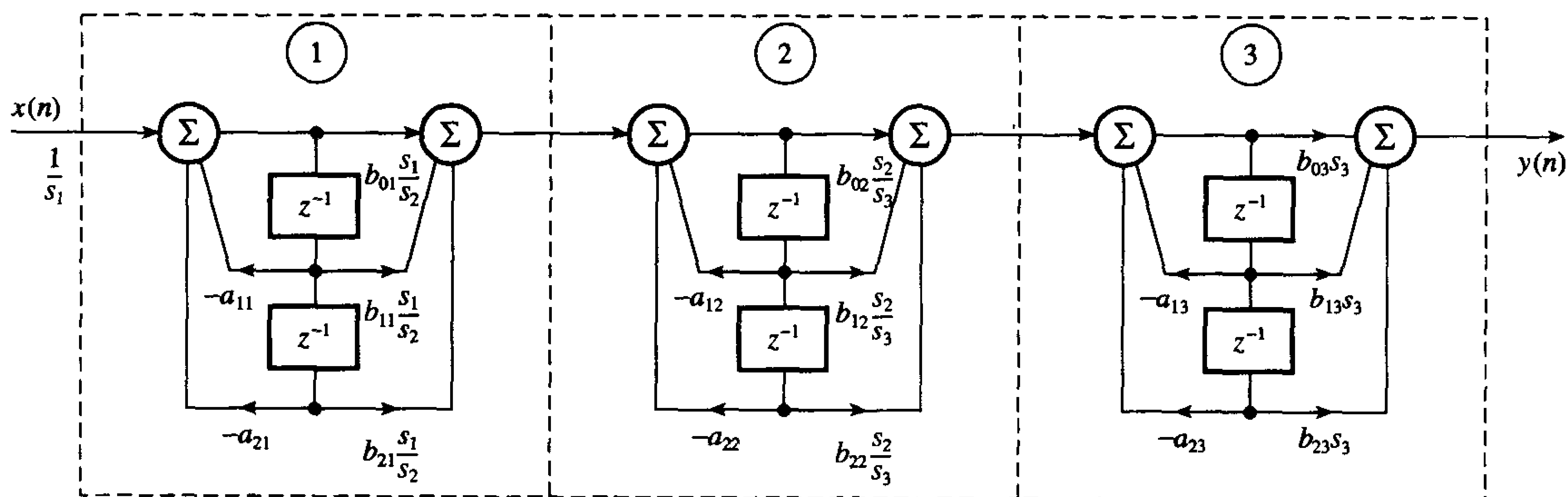


图13.12 对一个6阶IIR滤波器的串联实现进行伸缩变换（将伸缩因子放入分子或前馈系数中）

例13.11 对具有下面传递函数的滤波器进行伸缩变换，比较3种方法的伸缩因子，假定采用二阶子滤波器的串联实现：

$$H(z) = H_1(z)H_2(z)H_3(z)$$

其中

$$H_1(z) = \frac{1 + 0.2189z^{-1} + z^{-2}}{1 - 0.0127z^{-1} + 0.9443z^{-2}}$$

$$H_2(z) = \frac{1 - 0.5291z^{-1} + z^{-2}}{1 - 0.1731z^{-1} + 0.7252z^{-2}}$$

$$H_3(z) = \frac{1 + 1.5947z^{-1} + z^{-2}}{1 - 0.6152z^{-1} + 0.2581z^{-2}}$$

解:

使用 FWA 程序，分别获得 3 种方法的伸缩因子 s_1 到 s_3，列表如下：

	L_1	L_2	L_∞
s_1	20.9608	3.0388	13.4098
s_2	19.0361	2.5358	10.1366
s_3	14.4467	2.9146	6.4087

如前面所说，L_1 范数型永远是最大的，而 L_2 则永远是最小的。

13.4.7　并行实现中的伸缩变换

图 13.13 描述了在一个并行实现的 6 阶 IIR 滤波器中的伸缩变换机制。可以看到，每个二阶子滤波器都像前面讨论的那样单独进行伸缩变换。在每个子滤波器输入处的伸缩因子 s_i 保证了在对应节点 $w_i(n)$ 处不会产生溢出。为保证子滤波器的增益在伸缩变换前后不变，在流图中将前馈系数 b_{ki} 乘上 s_i。伸缩因子由下式给出：

$$s_i = \|F_i(z)\|_p$$

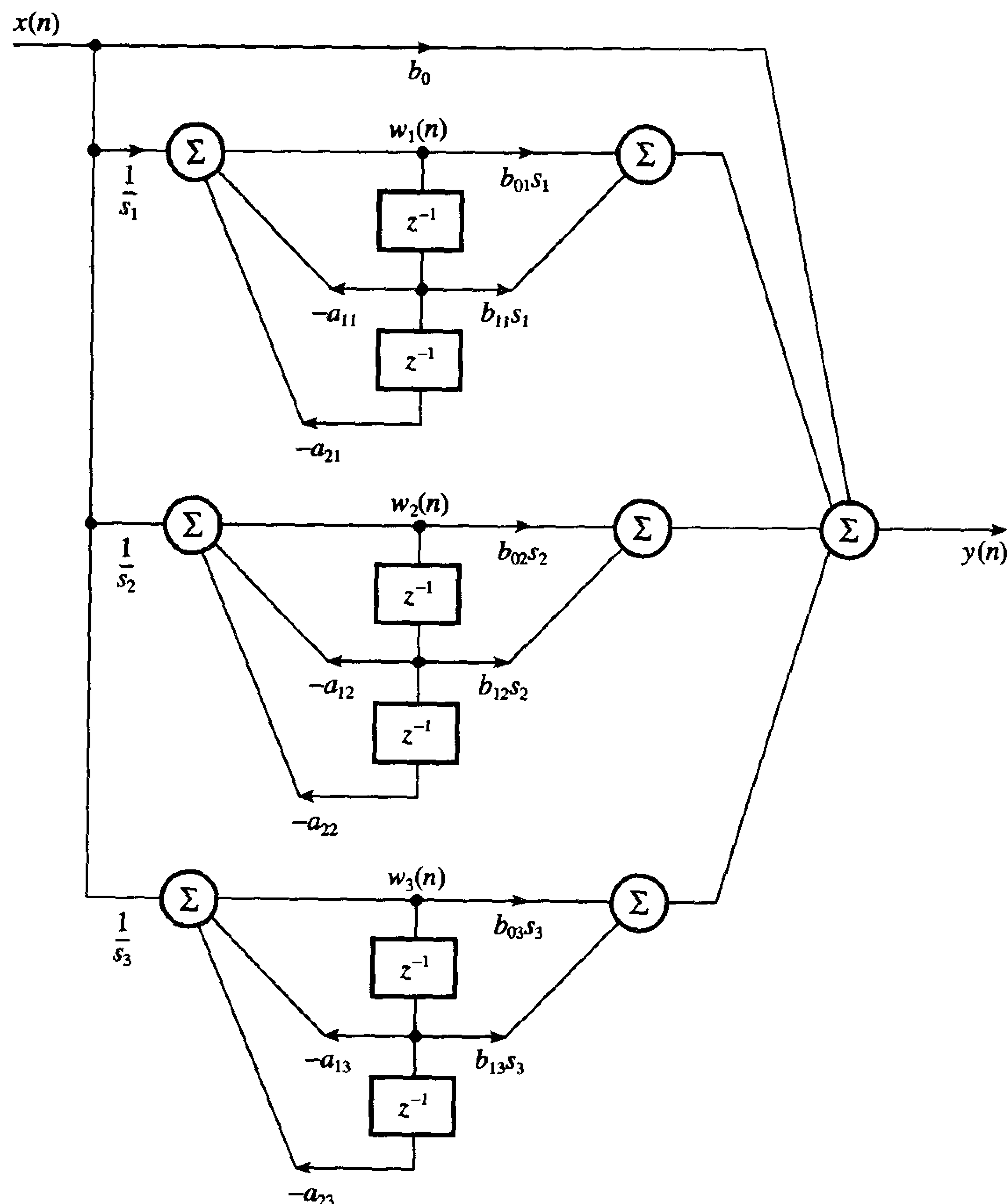


图 13.13　一个并行实现 6 阶 IIR 滤波器中的伸缩变换

其中 $F_i(z)$是从输入 $x(n)$到节点 $w_i(n)$的传递函数，表示如下：

$$F_i(z)=\frac{1}{1+a_{1i}z^{-1}+a_{2i}z^{-2}},\quad i=1,2,3$$

例 13.12 采用三种伸缩变换方法对下面传递函数所确定的滤波器进行伸缩变换，比较它们的伸缩因子：

$$H(z)=\frac{1.2916-0.08407z^{-1}}{1-0.131z^{-1}+0.3355z^{-2}}+\frac{7.5268}{1-0.049z^{-1}}-8.6788$$

解：

利用 FWA 程序，计算三种方法的伸缩因子，列表如下。

	L_1	L_2	L_∞
s_1	1.7345	1.0667	1.5126
s_2	1.0515	1.0012	1.0515

13.4.8 输出溢出的检测和防止

如果使用L_2和L_∞范数型，那么输出溢出是有可能的，虽然只是偶尔出现。在这种情况下，通常的办法是在滤波器输出端采用饱和运算（规则）。基本上，当输出溢出时，根据真实数据抽样的符号，将数据设定为容许的最大正值或负值。结果显示这种方法对最终输出的溢出很有效。如果输出不饱和，那么将是错误的，会导致不希望的后果，例如在数字音响中出现讨厌的声音。在标准结构中，都是这种情况。在直接结构中，如果最终输出的溢出没有得到校正，它会反馈到乘法器，影响接下来的输出抽样。

图 13.14(a)和图 13.14(b)分别给出了 2 的补码运算与饱和算术的溢出检测特性。在图中，y是正确的输出，$\hat{y}$是溢出输出。在现代 DSP 处理器中，一个趋势是加法器中提供额外的保护位，防止或降低出现溢出错误的可能性。例如在 DSP56300 中，一个 56 位的加法器包含了 8 个保护位。因此，它在计算中每隔 256 次溢出才会产生一个真正的溢出。

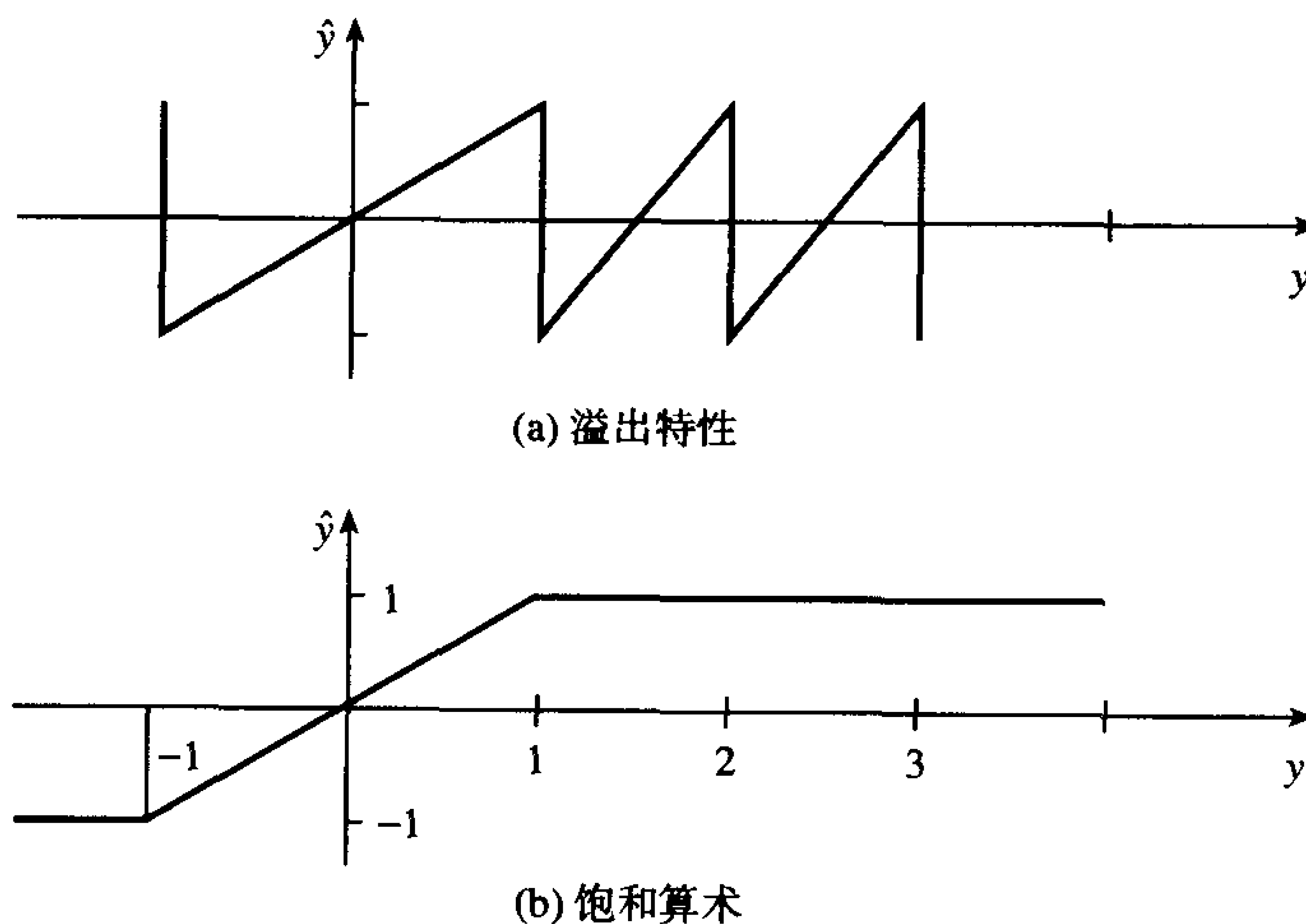


图 13.14 2 的补码运算与饱和算术的溢出检测特性

13.4.9 IIR 数字滤波器中的乘法舍入误差

乘法舍入误差分析是一个扩展的问题。我们在这里只给出简要的分析，目的在于使你意识到误差的实质、它们的后果以及在需要的时候怎样消除它们。

在 IIR 滤波中基本的操作是下面熟悉的二阶差分方程：

$$y(n)=\sum_{k=0}^{2}b_k x(n-k)-\sum_{k=1}^{2}a_k y(n-k) \tag{13.18}$$

其中 $x(n-k)$和 $y(n-k)$分别是输入和输出数据抽样，b_k 和 a_k 则是滤波器系数。在实践中，这些变量经常用定点数来表示。通常来说，乘积项 $b_k x(n-k)$或 $a_k y(n-k)$需要比每个运算数更多的位来表示。例如，一个 B 位的数据与一个 B 位的系数相乘，乘积有 $2B$ 位长。对于递归系统，如果不将结果数的长度缩短，不断进行的运算会导致位数不受限制地增长。

截断或舍入操作用于将结果量化回容许的长度。乘积的量化导致误差，通常称为数据的舍入误差，使输出 SNR 降低。这些误差还可能导致数字滤波器输出的小幅振荡，即便滤波器没有输入也是如此。

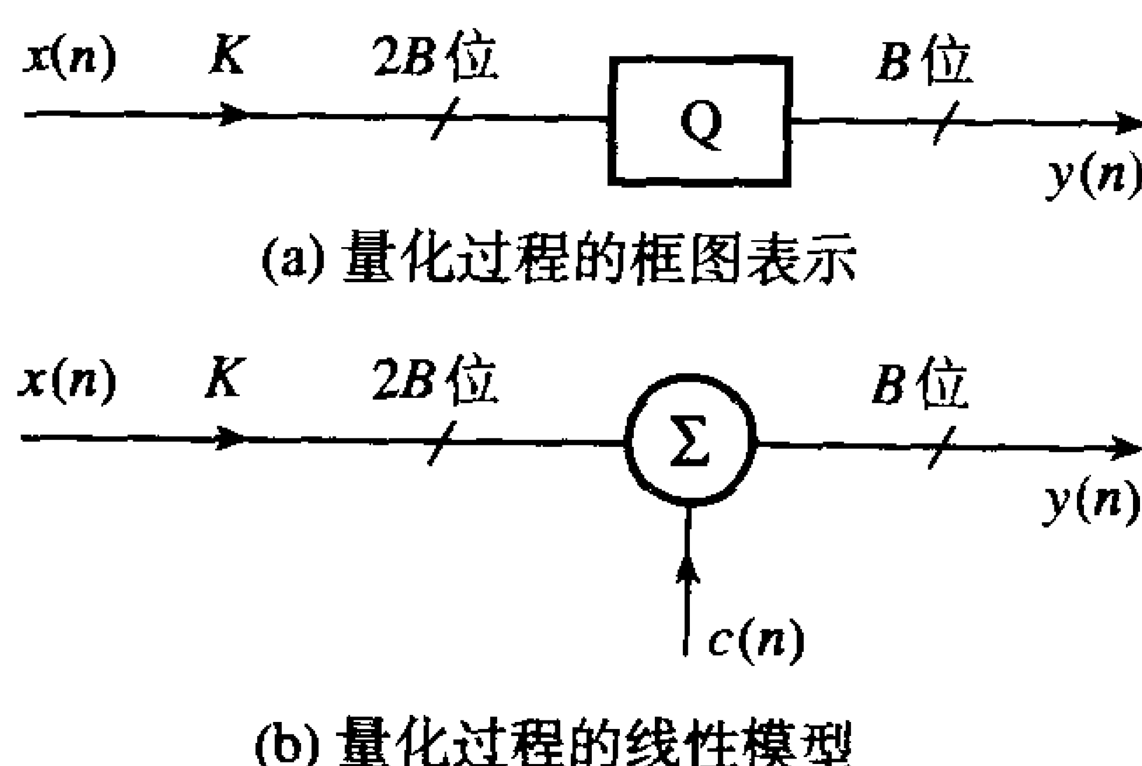
图 13.15(a)给出了一个乘积量化的框图表示，而图 13.15(b)则给出了乘积量化效果的线性模型。模型包含了一个理想乘法器，具有无限的精度，串联一个加法器，与噪声抽样 $e(n)$相加，代表在乘积量化过程中的误差。为了简便起见，这里我们假定 $x(n)$、$y(n)$和 K 每个都用 B 位表示。所以

$$y(n)=Kx(n)+e(n) \tag{13.19}$$

由于乘积量化所带来的噪声功率为

$$\sigma_r^2=\frac{q^2}{12}$$

这里 r 表示舍入误差，q 是由字长所确定的量化阶梯。舍入噪声被认为是具有零均值和恒定方差的随机变量。尽管这种假设并不总是成立的（例如对窄带、低水平信号的情况），但对评估滤波器性能是很有用的。



(a) 量化过程的框图表示

(b) 量化过程的线性模型

图 13.15　乘积量化误差的表示

在一个 DSP 系统中，舍入噪声可能会进入接下来的级，然后被放大、衰减或改变。舍入误差所产生的总输出噪声取决于系统的实现结构。当滤波器采用串联实现时，一个子滤波器产生的噪声进入下一个子滤波器。因此应该安排好各个子滤波器的先后顺序，使得舍入误差产生的总输出噪声最小。

13.4.10　舍入误差对滤波器性能的影响

舍入误差对滤波器性能的影响取决于所使用的滤波器结构和在何处对结构进行量化。图 13.16(a)显示了采用早先介绍的直接型实现的量化噪声模型。在图中假定输入数据 $x(n)$、输出数据 $y(n)$和滤波器系数都用 B 位数来表示（包括符号位）。乘法后通过舍入（或截断）将乘积量化回 B 位。

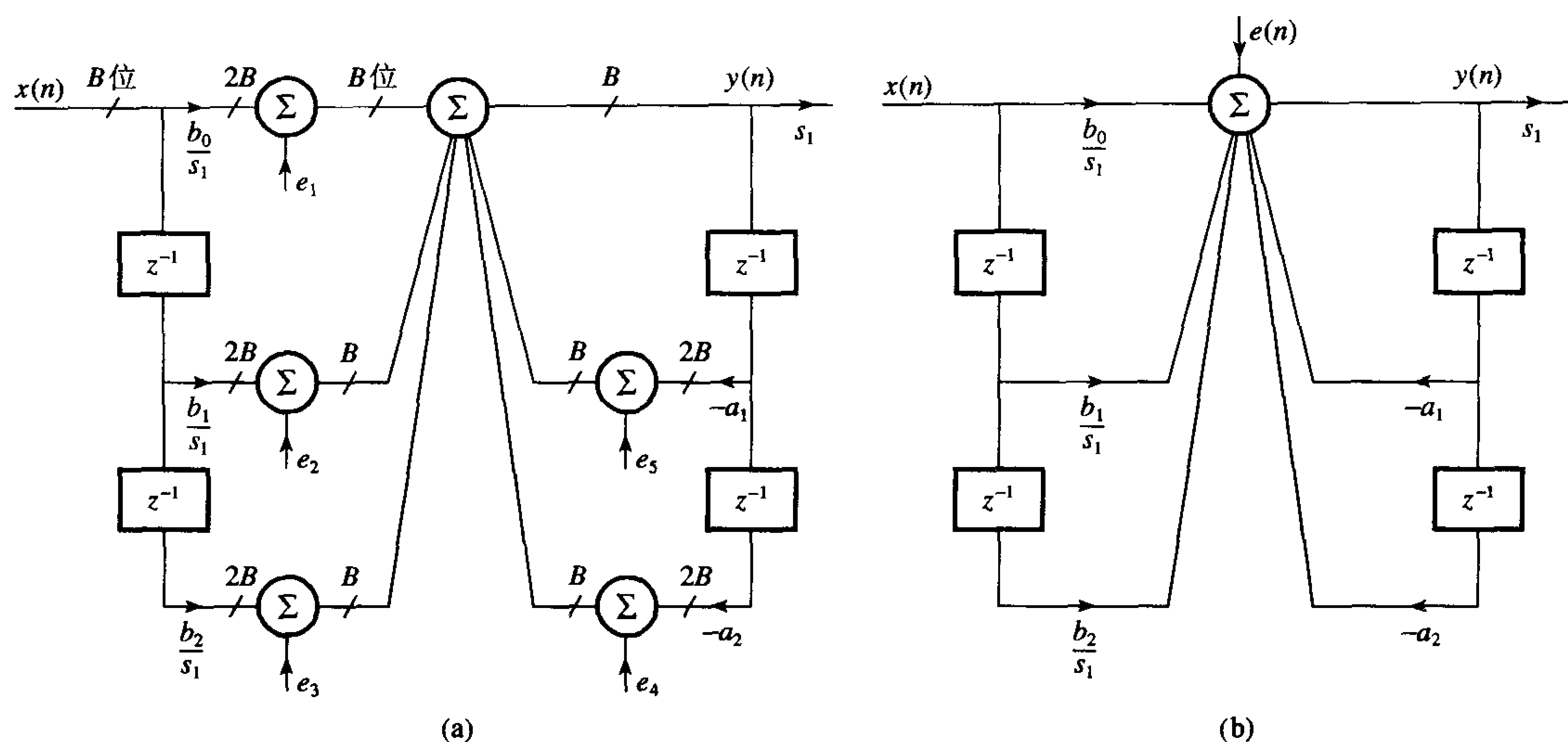


图 13.16 直接型实现的子滤波器的乘积量化噪声模型。在(a)中所有噪声源由于加到同一点而在(b)中被合并

由于图 13.16(a)中所有的噪声源 e_1 到 e_5 都进入同一个点（即中间的加法器），总的输出噪声功率是单个噪声功率的和（参见图 13.16(b)）：

$$\sigma_{\text{or}}^2 = \frac{5q^2}{12}\left[\frac{1}{2\pi \mathrm{j}}\oint_{\mathrm{c}} F(z)F(z^{-1})\frac{\mathrm{d}z}{z}\right]s_1^2$$

$$= \frac{5q^2}{12}\left[\sum_{k=0}^{\infty} f^2(k)\right]s_1^2 \tag{13.20}$$

$$= \frac{5q^2}{12}\|F(z)\|_2^2 s_1^2 \tag{13.21}$$

其中

$$F(z) = \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

$$f(k) = Z^{-1}[F(z)]$$

是 $F(z)$ 的 z 反变换，是从各个噪声源到滤波器输出的冲激响应，$\|\cdot\|_2^2$ 是 L_2 型范数，$q^2/12$ 是固有的乘积舍入噪声功率。在滤波器输出端的总噪声功率是乘积舍入噪声和ADC量化噪声的和（参见13.8式和 13.20 式）：

$$\sigma_{\text{o}}^2 = \sigma_{\text{oA}}^2 + \sigma_{\text{or}}^2$$

$$= \frac{q^2}{12}\left[\sum_{k=0}^{\infty} h^2(k) + 5s_1^2\sum_{k=0}^{\infty} f^2(k)\right] \tag{13.22}$$

$$= \frac{q^2}{12}[\|H(z)\|_2^2 + 5s_1^2\|F(z)\|_2^2]$$

对于标准型子滤波器，如图 13.17(a)，噪声模型又包含了一个伸缩因子，而它同样产生了一个舍入误差。噪声源 $e_1(n)$ 到 $e_3(n)$ 都进入左侧的加法器，而噪声源 $e_4(n)$ 到 $e_6(n)$ 则直接从滤波器输出。对进入同一个点的噪声源进行合并，产生了如图 13.17(b)所示的噪声模型。假定噪声源之间是无关的，总的噪声贡献是各自噪声贡献的简单叠加：

$$\sigma_{\text{or}}^2 = \frac{3q^2}{12}\sum_{k=0}^{\infty} f^2(k) + \frac{3q^2}{12} = \frac{3q^2}{12}[\|F(z)\|_2^2 + 1] \tag{13.23}$$

其中 $f(k)$ 是从噪声源 e_1 到滤波器输出的冲激响应，$F(z)$ 是相应的传递函数，即

$$F(z) = s_1\frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} = s_1 H(z) \tag{13.24}$$

滤波器输出的总噪声（ADC+ 舍入噪声）为

$$\begin{aligned}\sigma_{\text{o}}^2 &= \sigma_{\text{oA}}^2 + \sigma_{\text{or}}^2 \\ &= \frac{q^2}{12}\left\{3\left[1 + s_1^2\sum_{k=0}^{\infty} h^2(k)\right] + \sum_{k=0}^{\infty} h^2(k)\right\} \\ &= \frac{q^2}{12}\{3[1 + s_1^2\|H(z)\|_2^2] + \|H(z)\|_2^2\}\end{aligned} \tag{13.25}$$

应比较 13.25 式与 13.22 式，注意到伸缩因子的引入增加了输出噪声。

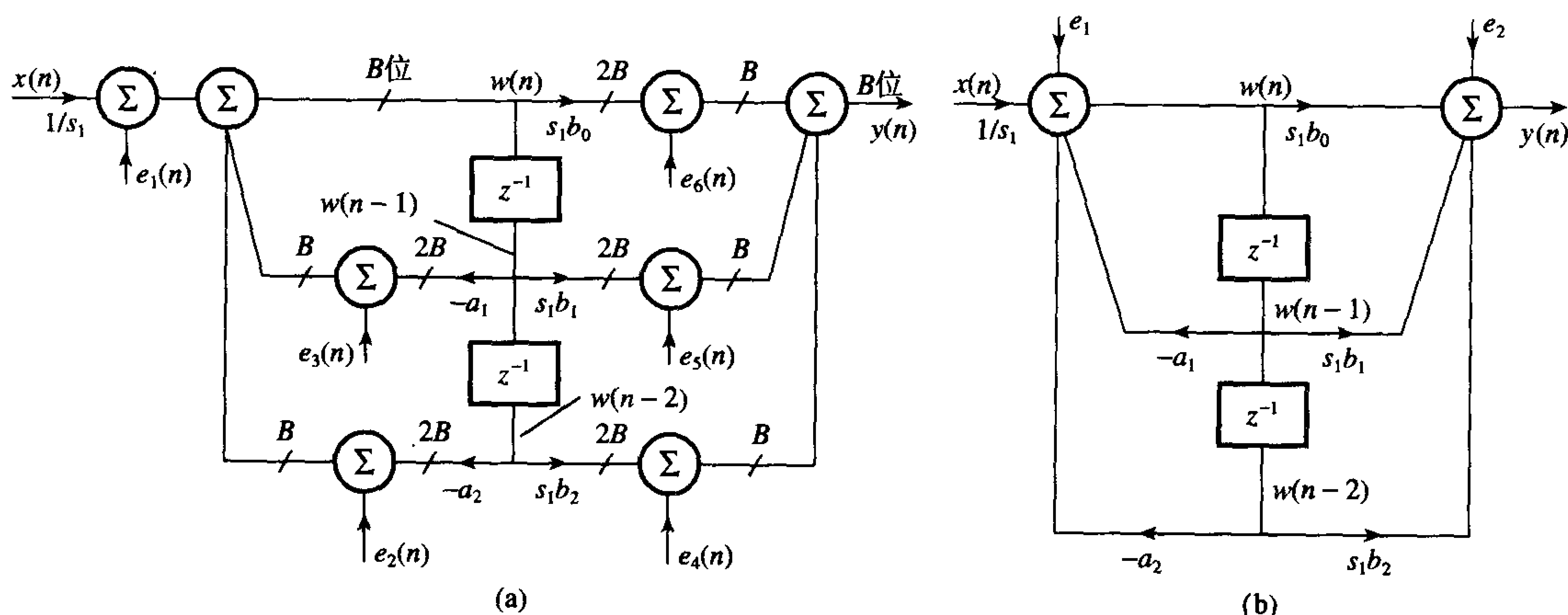


图 13.17　标准型子滤波器的乘积量化噪声模型。在(a)中加到同一点的噪声源在(b)中被合并

例 13.13　一个具有如下传递函数特性的滤波器：

$$H(z) = \frac{0.1436 + 0.2872z^{-1} + 0.1436z^{-2}}{1 - 1.8353z^{-1} + 0.9747z^{-2}}$$

滤波器用一个 8 位系统来实现，输入数据 $x(n)$，输出数据 $y(n)$，滤波器系数用 8 位定点小数和 2 的补码来表示。

假定使用二阶标准型子滤波器来实现该滤波器，每次乘法后乘积（16 位表示）立刻被量化成 8 位。

(1) (a) 给出实现草图，显示滤波器内的舍入误差源，为系统确定合适的伸缩因子；

(b) 估计由于舍入噪声误差而产生的总稳态噪声输出功率，以及相应的输出 SNR 下降，用分贝数表示；

(2) 如果使用直接型结构来实现滤波器，重复第(1)个问题。

解：

(1) (a) 带噪声源的系统实现图如图 13.17(b)所示。噪声源 e_1 表示由于将每个乘积 $a_1w(n-1)$、$a_2w(n-2)$ 和 $x(n)/s_1$ 从 16 位舍入（或截断）到 8 位而产生的误差的和。噪声源 e_2 则是右侧加法器的三个 16 位输入被量化成 8 位所产生的。

利用 FWA 程序，三种范数下量化因子分别为 $s_1 = 133.899\,66(L_1)$、$s_1 = 12.1395(L_2)$ 和 $s_1 = 102.088(L_\infty)$。

(b) 利用在 13.23 式基础上的有限字长分析程序，舍入噪声产生的输出噪声功率为

$$\sigma_{or}^2 = 1668.03q^2$$

这里我们假定按 L_2 范数进行伸缩变换。

根据 13.8 式，由于 ADC 而产生的输出噪声估计为 $3.7724q^2$。因此总的输出噪声功率为

$$\sigma_o^2 = (1668.03 + 3.7724)q^2 = 1671.8024q^2$$

假设是一个随机输入信号，输出信号功率为

$$\sigma_x^2 = \frac{1}{3}\|H(z)\|_2^2 = 15.0896$$

SNR（不含舍入误差）为

$$\frac{15.0896}{3.7724q^2} = \frac{4}{q^2}$$

SNR（包含舍入误差）为

$$\frac{15.0896}{1671.8024q^2}$$

由于舍入误差导致的 SNR 下降为

$$10\log\left(\frac{4/q^2}{9.0260\times10^{-3}/q^2}\right) = 26.47\,\text{dB}$$

(2) 对于直接型实现，舍入误差产生的总输出噪声功率为 $9048.82q^2$。L_2 范数下，由于舍入误差导致的 SNR 下降为 33.8 dB。如果不带伸缩变换，SNR 只下降约 1.11 dB。当不带伸缩变换时 SNR 下降如此之小，因此在一些应用里，直接型实现可以避免伸缩变换而更受欢迎（Dattorro, 1988）。

13.4.11 串联和并联实现中的舍入噪声

13.4.11.1 串联

图 13.18 绘出了由二阶标准型子滤波器串联实现的六阶 IIR 系统，其中噪声源如前面所建议的进行了合并，为了简便而重新编号。因此 e_1 是三个乘法器输出到左侧加法器时（量化）所产生噪声源的和。复合噪声源 e_1 将要经过三个子滤波器 $H_1(z)$、$H_2(z)$ 和 $H_3(z)$，而复合噪声源 e_2 则经过传递函数 $H_2(z)$ 和 $H_3(z)$，等等。

舍入误差产生的总输出噪声是所有 6 个噪声源的和：

$$\begin{aligned}\sigma_{or}^2 &= \frac{3q^2}{12}\sum_{k=0}^{\infty}f_1^2(k) + \frac{3q^2}{12}\sum_{k=0}^{\infty}f_2^2(k) + \frac{2q^2}{12}\sum_{k=0}^{\infty}f_3^2(k) + \frac{3q^2}{12}\sum_{k=0}^{\infty}f_4^2(k) \\ &\quad + \frac{2q^2}{12}\sum_{k=0}^{\infty}f_5^2(k) + \frac{3q^2}{12} \\ &= \frac{q^2}{12}\left[3\sum_{k=0}^{\infty}f_1^2(k) + 5\sum_{k=0}^{\infty}f_3^2(k) + 5\sum_{k=0}^{\infty}f_5^2(k) + 3\right] \qquad (13.26) \\ &= \frac{q^2}{12}[3\|F_1(z)\|_2^2 + 5\|F_3(z)\|_2^2 + 5\|F_5(z)\|_2^2 + 3]\end{aligned}$$

其中$f_i(k)$是噪声源e_i与输出之间的冲激响应。由e_2和e_3（参见图 13.18）所产生的噪声分量都经过了同样的子滤波器，即$H_2(z)$和$H_3(z)$，所以它们对输出的贡献被合并了。同样处理对e_4和e_5的噪声分量贡献。

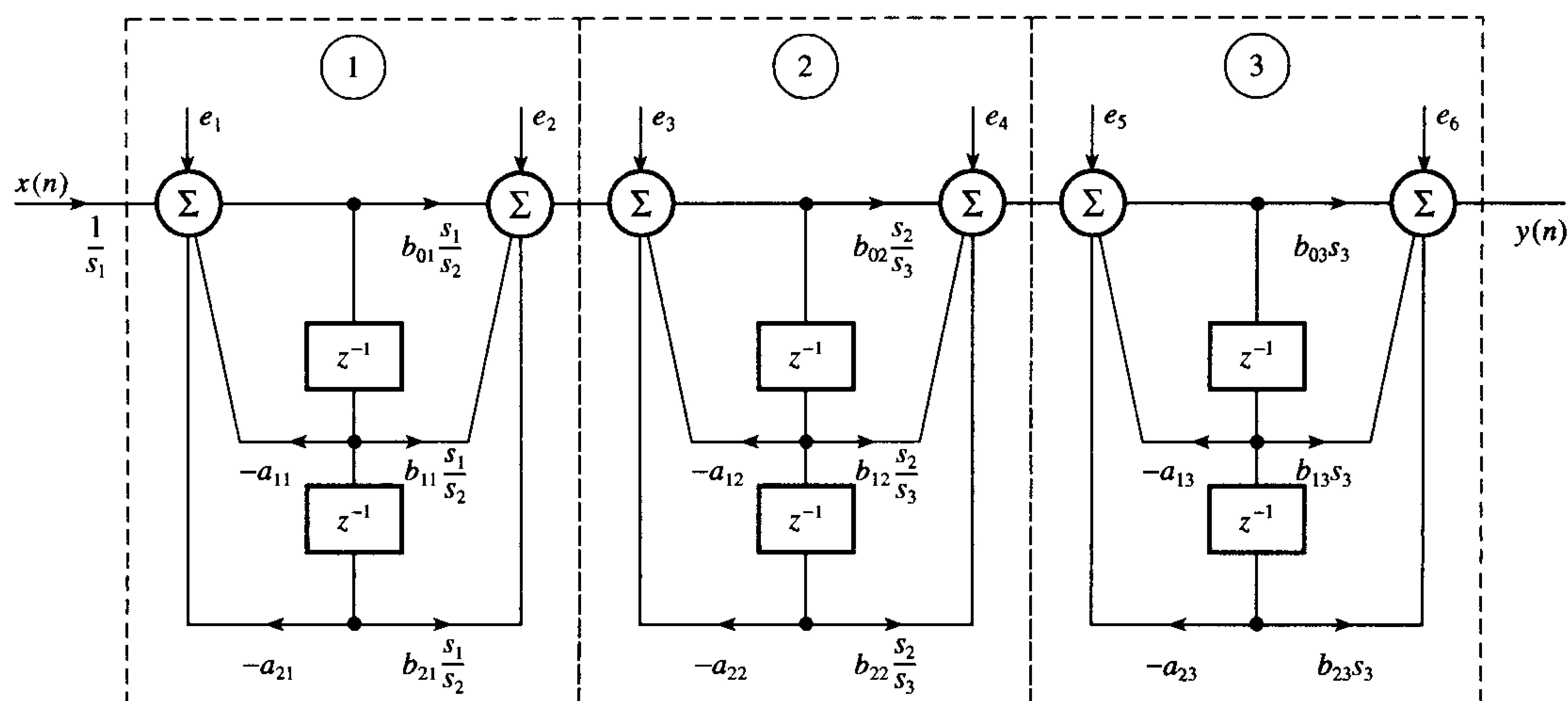


图 13.18　串联实现的六阶 IIR 滤波器的噪声模型

13.4.11.2　并联

图 13.19 给出了并联实现的六阶 IIR 系统的舍入噪声模型。如前面所述，单个乘积量化所产生的噪声源被合并。噪声源e_1到e_3每个都经过一个子滤波器到达输出，而剩余的噪声源则直接输出。从e_1到e_3，每个噪声源对输出噪声的贡献为

$$\begin{aligned}\sigma_{\mathrm{r},i}^2 &= \frac{3q^2}{12}\sum_{k=0}^{\infty} f_i^2(k) = \frac{3q^2}{12}\|F_i(z)\|_2^2, \quad i=1,2,3 \\ &= \frac{3q^2}{12}s_i^2\sum_{k=0}^{\infty} h_i^2(k) = \frac{3q^2}{12}s_i^2\|H_i(z)\|_2^2\end{aligned} \tag{13.27}$$

其中$H_i(z)$和$h_i(k)$分别是子滤波器i的传递函数和相应的冲激响应。$F_i(z)$和$f_i(k)$则是噪声源i经过的传递函数和相应的冲激响应。

噪声源e_i（$i=4,5,6$）都直接进入输出而产生了e_7。因此总的输出噪声功率为

$$\begin{aligned}\sigma_{\mathrm{or}}^2 &= \frac{q^2}{12}\left\{7 + 3\sum_{i=1}^{3}\left[s_i^2\sum_{k=0}^{\infty} h_i^2(k)\right]\right\} \\ &= \frac{q^2}{12}\left[7 + 3\sum_{i=1}^{3} s_i^2\|H_i(z)\|_2^2\right]\end{aligned} \tag{13.28}$$

通常来说，对L个子滤波器的并联实现，由于舍入误差而产生的输出功率为

$$\sigma_{\mathrm{or}}^2 = \frac{q^2}{12}\left[2L + 1 + 3\sum_{i=1}^{L} s_i^2\|H_i(z)\|_2^2\right] \tag{13.29}$$

对所有上面公式给出的舍入噪声功率的估计可以通过一个合适的计算机程序而方便地获得（例如，早先介绍的有限字长分析程序）。

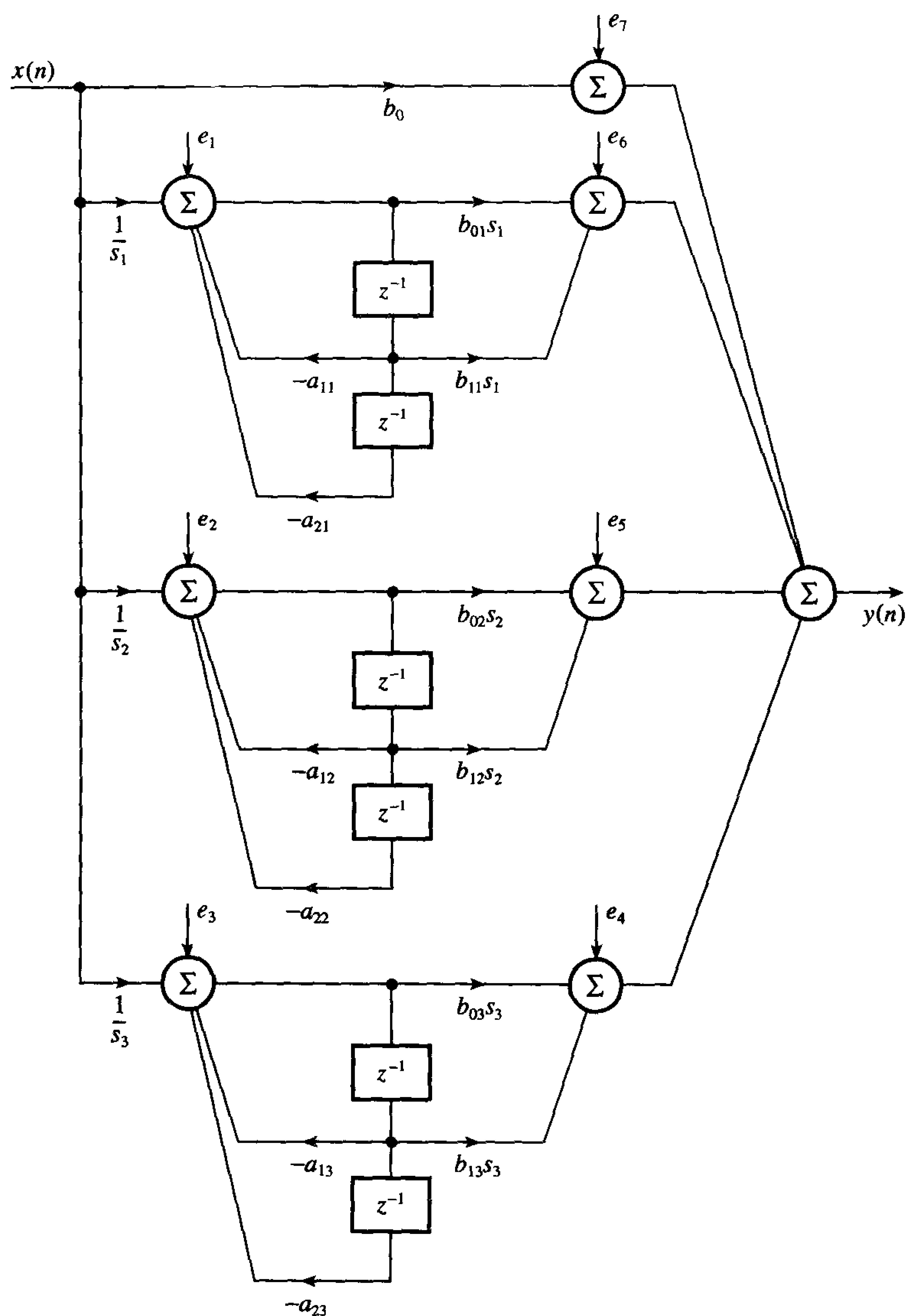


图 13.19 并联实现的六阶 IIR 滤波器的噪声模型

例 13.14 下面的传递函数代表一个四阶 IIR 滤波器（Mitra et al., 1974）:

$$H(z)=\frac{1-2z^{-1}+z^{-2}}{1+0.777z^{-1}+0.3434z^{-2}}\frac{1-0.707z^{-1}+z^{-2}}{1+0.01877z^{-1}+0.801z^{-2}}0.093226$$

估计 SNR 下降的分贝数，如果滤波器的实现方式为

(1) 2 个二阶子滤波器的串联，顺序与 $H(z)$的相同；

(2) 2 个二阶子滤波器的并联。

假定每种情况下乘积在做加法前被量化成 8 位。

解:

串联和并联实现结构的噪声模型分别同图13.18和图13.19完全一致(如果我们不考虑第三级)。

(1) 对于串联实现，L_1 伸缩因子（利用基于PC的程序）是

$$s_1 = 2.395\,746$$

$$s_2 = 15.703\,627$$

滤波器输出的舍入噪声功率、ADC噪声功率及信号功率分别是

$$\sigma_{\text{or}}^2 = 345.0391q^2$$

$$\sigma_{\text{oA}}^2 = 0.039\,76q^2$$

$$\sigma_{\text{y}}^2 = 0.1591$$

SNR（不含舍入误差）为

$$\frac{0.1591}{0.0397q^2}$$

SNR（包含舍入误差）为

$$\frac{0.1591}{345.0789q^2}$$

由于舍入误差导致的 SNR 下降为

$$10\log(345.0789/0.039\,76) \approx 39.4\text{ dB}$$

(2) 对于并联实现，利用部分分式展开（Mitra et al., 1974），传递函数变成

$$H(z) = 0.093\,326\left(1 + \frac{-5.162 + 0.7867z^{-1}}{1 + 0.777z^{-1} + 0.3434z^{-2}} + \frac{1.657\,36 + 0.2759z^{-1}}{1 + 0.018\,77z^{-1} + 0.801z^{-2}}\right)$$

滤波器输出的舍入噪声功率是

$$\sigma_{\text{or}}^2 = \frac{q^2}{12}\left[5 + 3\sum_{i=1}^{2} s_i^2 \|H_i(z)\|_2^2\right]$$

利用FWA程序，我们得到L_2的伸缩因子为$s_1 = 2.395\,746$，$s_2 = 5.450\,612$，$\|H_1(z)\|_2^2 = (7.378\,492)^2$，$\|H_2(z)\|_2^2 = (2.801\,937)^2$。因此我们有

$$\sigma_{\text{or}}^2 = \frac{q^2}{12}\{5 + 3[(2.395\,746)^2(7.378\,492)^2 + (5.450\,612)^2(2.801\,937)^2]\}$$

$$= 136.85q^2$$

这样，假定同样的输出信号和ADC噪声，在滤波器输出端的SNR（包含舍入误差）为

$$0.1591/(136.85 + 0.039\,76)q^2 = 1.162\,28 \times 10^{-3}/q^2$$

由于舍入误差导致的 SNR 下降变成

$$10\log[(136.85 + 0.039\,76)/0.039\,76] = 35.37\text{ dB}$$

13.4.12　乘积舍入噪声对现代DSP系统的影响

早期研究乘积舍入误差对滤波器性能的影响主要是基于单一的、固定的内部字长，且在做加法前强制性地将每个$2B$位乘积（严格地讲，为$2B$-1位）量化回B位。在现代DSP处理器中上述约束并不存在，因为它们支持双字长加法。所有现代DSP处理器的特征都至少具有一个内建（built-in）的16 × 16位乘法器和一个32位乘积寄存器，允许乘积按照32位数累加，即通常所称的16/32位结构。

图 13.20(a)给出了直接型二阶子滤波器的噪声模型，其量化是在乘积相加后进行的。在图中，乘积的和为 $2B$ 位的 $y'(n)$，被量化回 B 位。为了将这种方式与每个乘积分别量化的相区分，我们称之为累加后量化（post-accumulation quantization）。显然，在这种情况下量化只会产生一个噪声源。这时的输出噪声功率为

$$\sigma_{or}^2 = \frac{q^2}{12}\|F(z)\|_2^2 \tag{13.30}$$

其中

$$F(z) = \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

对于标准型二阶子滤波器的情况（参见图 13.20(b)），噪声源产生的输出噪声和相应的 SNR 为

$$\sigma_{or}^2 = \frac{q^2}{12}[s_1^2\|H(z)\|_2^2 + 1] \tag{13.31}$$

显然，对乘积进行加法后再舍入（参见 13.30 式和 13.31 式），与直接对每个乘积进行舍入相比，可以使舍入噪声显著减小。

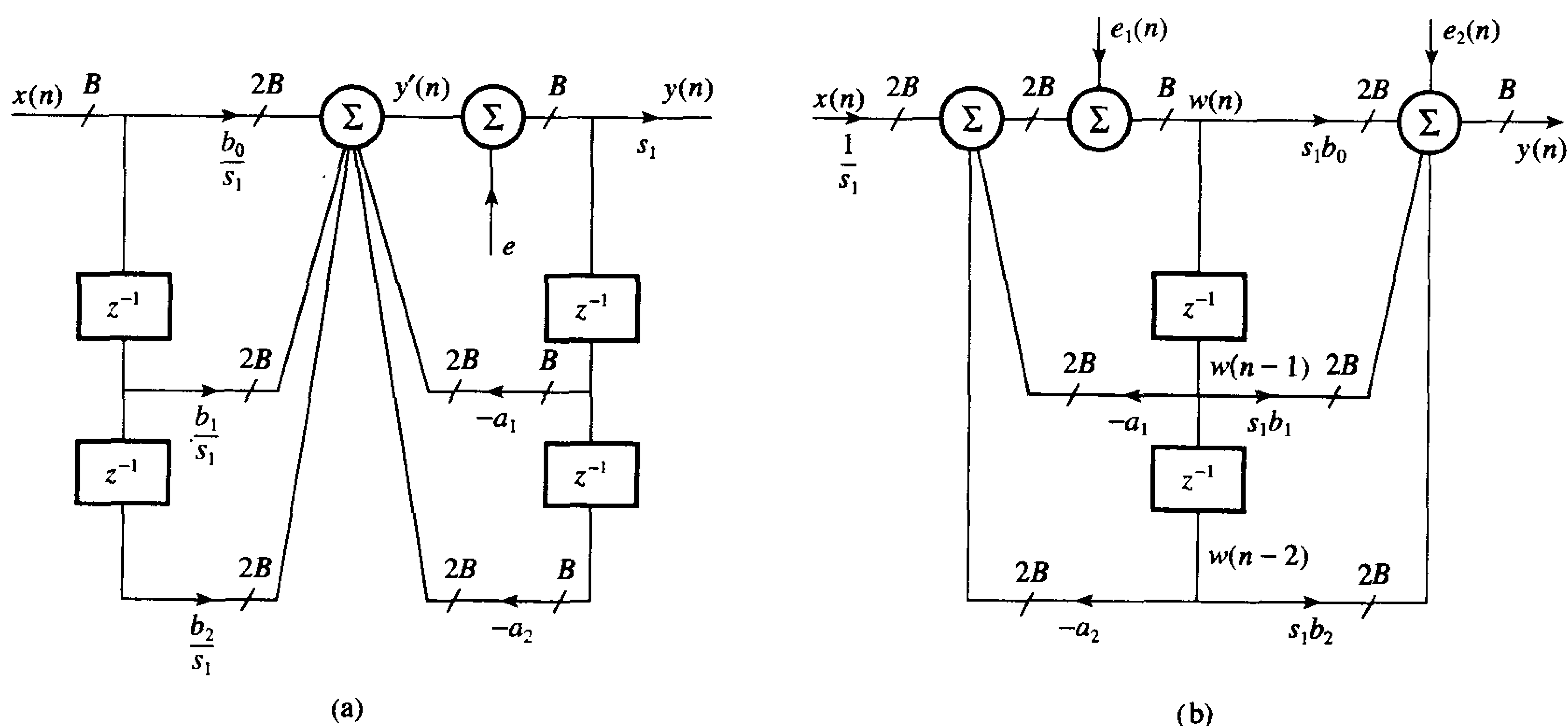


图 13.20 现代DSP系统中IIR子滤波器的噪声模型。滤波器内各不同点的字长被标注出来。假定输入数据和滤波器系数的字长都是 B 位

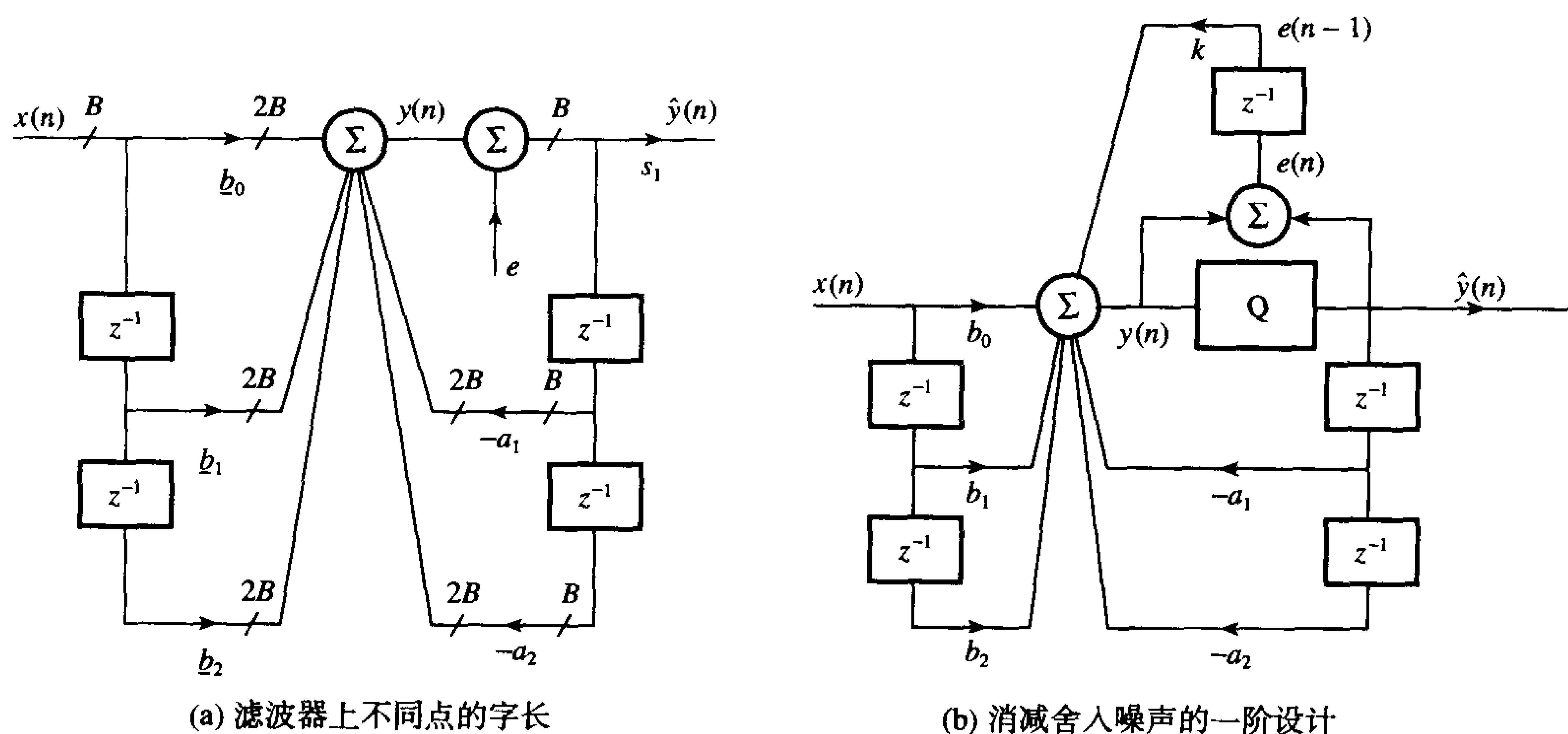
13.4.13 减少舍入噪声的方法

在实践中，滤波器内的某些点必须进行舍入或截断操作，以满足乘法器、数据存储器和与外部世界接口的字长要求。输入信号的电平较低时，对乘积的舍入或截断所导致的乘积舍入误差，使得滤波器输出产生了相当大的失真，这对于高保真系统（high fidelity system）应尽量予以消减。许多设计用于降低或消除IIR滤波器中的舍入误差影响。这些设计通过有效地改变噪声频谱的形状，降低或消除它们对特定频带滤波器的影响。所有这类设计统称为误差频谱整形（error spectral shaping, ESS）。

我们将介绍在直接I型二阶子滤波器上消减舍入误差的基本原则，如图 13.21(a)所示。在图中，滤波器参数（系数和数据）都用 B 位定点数表示，而加法器则有 $2B$ 位的宽度，即是一个 $B/2B$ 的实现方式。在现代 DSP 处理器中，B 的典型值是 16 位或 24 位。当加法器的输出被量化回 B 位时，产生了舍入噪声。可以看到量化输出的变换为

$$Y(z)=\frac{b_0+b_1z^{-1}+b_2z^{-2}}{1+a_1z^{-1}+a_2z^{-2}}X(z)-\frac{1}{1+a_1z^{-1}+a_2z^{-2}}E(z) \tag{13.32}$$

所以量化输出的频谱等于理想输出的频谱加上一个伸缩变换误差频谱。误差频谱被滤波器的极点 $1+a_1z^{-1}+a_2z^{-2}$ 所放大，而与滤波器特性关系不大。根据滤波器的类型，噪声可能在频率范围内的低端、中间或高端被放大。



(a) 滤波器上不同点的字长　　(b) 消减舍入噪声的一阶设计

图 13.21　直接 I 型子滤波器的舍入噪声

一个消减舍入噪声影响的一阶误差反馈设计在图 13.21(b)中给出。可以发现（参见例 13.16），在本例中带误差反馈的子滤波器的输出变换为

$$Y(z)=\frac{b_0+b_1z^{-1}+b_2z^{-2}}{1+a_1z^{-1}+a_2z^{-2}}X(z)-\frac{1-kz^{-1}}{1+a_1z^{-1}+a_2z^{-2}}E(z) \tag{13.33}$$

反馈系数 k 在误差频谱的通路中引入了一个零点，有效地抵消了滤波器极点的影响。图 13.22 揭示了误差反馈的零点对误差频谱的影响。显然，在没有误差反馈的情况下，误差频谱被滤波器的极点所放大。有了误差反馈系数，在误差通路中引入了一个零点，导致了噪声频谱的消减。对抗滤波器极点效应的恰当策略是，在反馈网络中尽可能地使零点与极点频率相同。

二阶噪声消减用于得到噪声频谱的更大程度消减，请参见图 13.23。

对直接和标准型子滤波器的通用噪声消减策略分别见图 13.24(a)和图 13.24(b)所示。在两个图中，反馈和前馈系数 a_i' 和 b_i'，用于改变舍入误差所经过的传递函数，在不影响希望信号的条件下，最小化输出的舍入噪声。在图 13.24(a)中，对左侧加法器输出的乘积和的量化产生了一个误差 $e_1(n)$，它是双精度变量 $y(n)$ 的低字部分。尽管它们都是 2B+1 位长，但是乘积 $a_1'e_1(n-1)$ 和 $a_2'e_2(n-2)$ 的权重与加法器的其他输入并不相同，所以它们必须在与其他输入相加前被重新排列或量化。这种情况下的量化误差表示为 $e_2(n)$。同样，$b_0'e(n)/s_1$、$b_1'e(n-1)/s_1$ 和 $b_2'e(n-2)/s_1$ 各项需要在与右侧加法器的其他输入相加前被量化，由此产生了误差项 $e_3(n)$。最后，右侧加法器的输出将从 2B+1 位量化回 B+1 位，从而产生误差项 $e_4(n)$。对于图 13.24(b)的标准型子滤波器，也要进行同样的考虑。

对于直接型实现 ESS（参见图 13.24(a)），输出噪声为

$$\sigma_{or}^2=\frac{q^2}{12}\left[\sum_{k=0}^{\infty}f_1^2(k)+\sum_{k=0}^{\infty}f_2^2(k)+2s_1^2\right] \tag{13.34}$$

其中 $f_1(k)$ 和 $f_2(k)$ 分别是从噪声源 1 和 2 到滤波器输出的冲激响应。

(a) 极零图：(i) 滤波器；(ii) $k = 1$的误差反馈网络

(b) $k = 0$的误差频谱

(c) $k = 1$的误差频谱

图 13.22　误差反馈系数的效果示意图。注意到误差反馈网络的零点位置尽可能地与滤波器极点位置相近

图 13.23　一个二阶噪声消减方案

对于标准型子滤波器（参见图 13.24(b)），舍入产生的输出噪声功率为

$$\begin{aligned}\sigma_{\text{or}}^2 &= \frac{q^2}{12}\left[\sum_{k=0}^{\infty} f_1^2(k) + \sum_{k=0}^{\infty} f_2^2(k) + 2\right] \\ &= \frac{q^2}{12}[\|F_1(z)\|_2^2 + \|F_2(z)\|_2^2 + 2]\end{aligned} \tag{13.35}$$

其中

$$F_1(z) = (b_0' + b_1'z^{-1} + b_2'z^{-2})s_1 + \frac{(1 + a_1'z^{-1} + a_2'z^{-2})(b_0 + b_1z^{-1} + b_2z^{-2})s_1}{1 + a_1z^{-1} + a_2z^{-2}}$$

$$F_2(z) = \frac{(b_0 + b_1z^{-1} + b_2z^{-2})s_1}{1 + a_1z^{-1} + a_2z^{-2}}$$

误差频谱整形（ESS）系数 a_i' 和 b_i' 的选择，决定了噪声消减方案的有效性。在实践中，ESS 的系数通常限于整数，以避免更多的量化过程。一阶和二阶 ESS 在实际应用中最常使用。

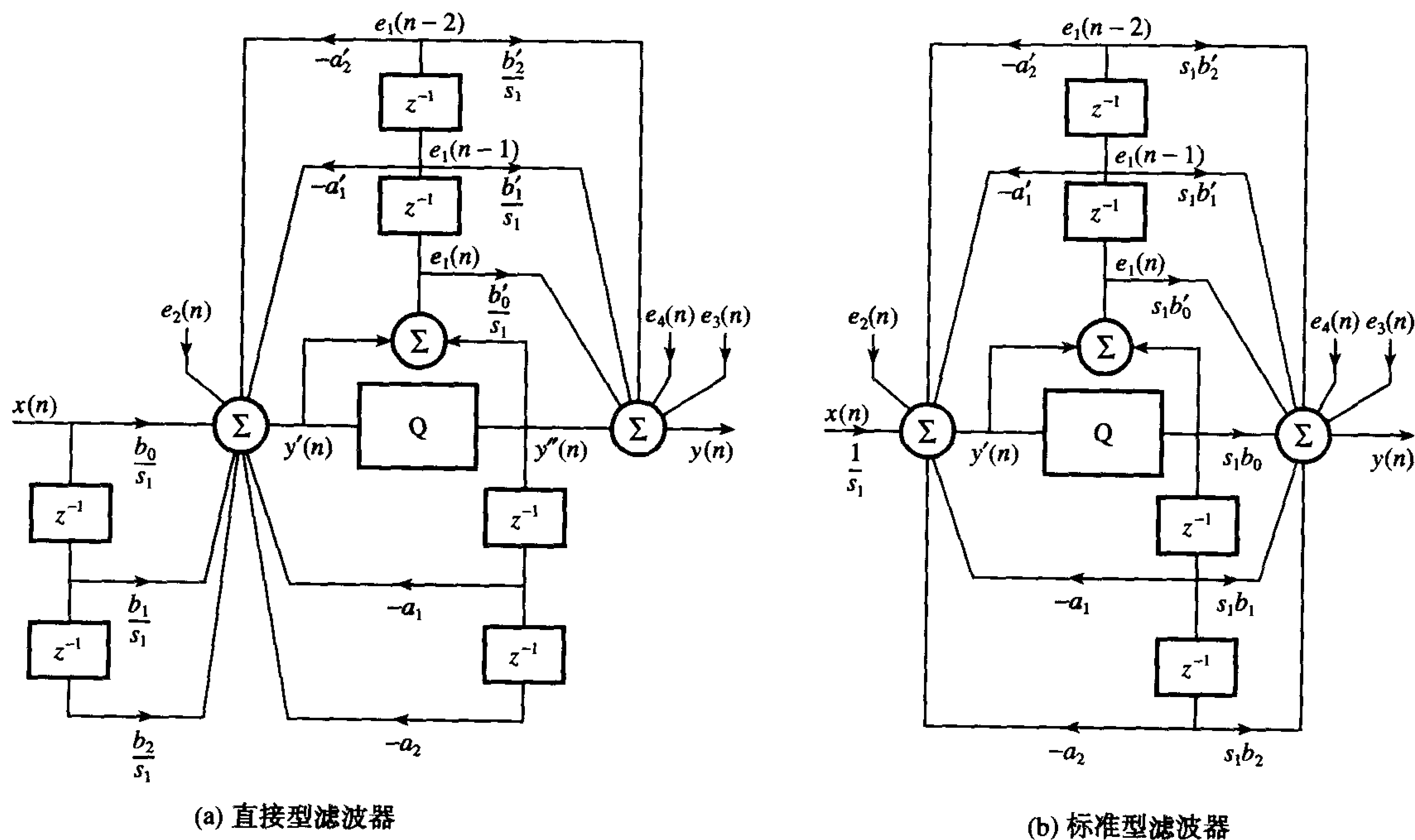


图 13.24　通用噪声消减方案

在一阶噪声消减方案中，误差前馈系数被设定为零，一个误差反馈系数被设定为整数。一阶噪声消减方案对于窄带低通或高通滤波器的舍入噪声消减特别有效，因为它能够在舍入误差的通道上放置一个单零点，其频率恰在频带的低端或高端。该方案的好处是，滤波器增加的计算复杂度是适度的。

最优 ESS 可以通过二阶方案来实现，这时滤波器输出端的舍入噪声影响可以完全消除。对于直接型子滤波器（参见图 13.24(a)），最优 ESS 的设置为

$$b_i' = 0,\ i = 0, 1, 2;\quad a_i' = a_i,\ i = 1, 2 \tag{13.36}$$

这时，输出舍入噪声大量消减，只剩固有的舍入噪声。输出舍入噪声功率消减为

$$\sigma_{\text{or}}^2 = \frac{q^2}{12}\left[1 + \sum_{k=0}^{\infty} f^2(k)\right]$$
$$= \frac{q^2}{12}[1 + \|F(z)\|_2^2] \tag{13.37}$$

其中

$$F(z) = \frac{s_1}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

对于标准型子滤波器，其最优设置为

$$b_i' = -b_i, i = 0, 1, 2; \quad a_i' = a_i, i = 1, 2 \tag{13.38}$$

输出噪声功率为

$$\sigma_{\text{or}}^2 = \frac{q^4}{12}\sum_{k=0}^{\infty} f^2(k) = \frac{q^4}{12}\|F(z)\|_2^2$$

其中

$$F(z) = \frac{(b_0 + b_1 z^{-1} + b_2 z^{-2})s_1}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

最优方案需要较多的计算量，并且如Mullis and Roberts(1982)所指出，在内部滤波器变量用双精度表示时才有效。除了前面提到的整数法，人们还提出了许多次优的其他方案（Higgins and Munson, 1982）。

例 13.15 比较具有下面传递函数特性的一个二阶 IIR 滤波器的舍入噪声性能：

$$H(z) = \frac{0.1436(1 + 2z^{-1} + z^{-2})}{1 - 1.8353z^{-1} + 0.9748z^{-2}}$$

如果滤波器用一个(a)标准型和(b)直接型滤波器来实现，且系数设置为

(1) $a_i' = 0, i = 1, 2$（没有误差反馈）
(2) $a_1' = -1, a_2' = 0$
(3) $a_1' = -2, a_2' = 0$
(4) $a_1' = -1, a_2' = 1$
(5) $a_1' = -2, a_2' = 1$

假定每种情况下所有前馈误差系数都为零。

解:

ESS 的实现结构如图 13.25 所示。标准型和直接型实现结构的舍入噪声的输出功率分别是

$$\sigma_{\text{or}}^2 = \frac{q^2}{12}[\|F_1(z)\|_2^2 + 1]$$

和

$$\sigma_{\text{or}}^2 = \frac{q^2}{12}[\|F_2(z)\|_2^2 + 1]$$

其中

$$F_1(z) = (1 + a_1'z^{-1} + a_2'z^{-2})\frac{(b_0 + b_1z^{-1} + b_2z^{-2})s_1}{1 + a_1z^{-1} + a_2z^{-2}}, \quad s_1 = 12.1395\text{（}L_2\text{ 比例）}$$

$$F_2(z) = \frac{(1 + a_1'z^{-1} + a_2'z^{-2})s_1}{1 + a_1z^{-1} + a_2z^{-2}}, \quad s_1 = 6.7282\text{（}L_2\text{ 比例）}$$

利用 FWA 程序，计算出每种情况下的输出噪声功率，列表于表 13.5。注意到对于第 3 种选择（$a_1' = -2,\ a_2' = 0$），输出噪声没有降低，反而增大了，这说明选择反馈系数的重要性。对于一阶方案，第 2 种选择对降低输出噪声最有效。

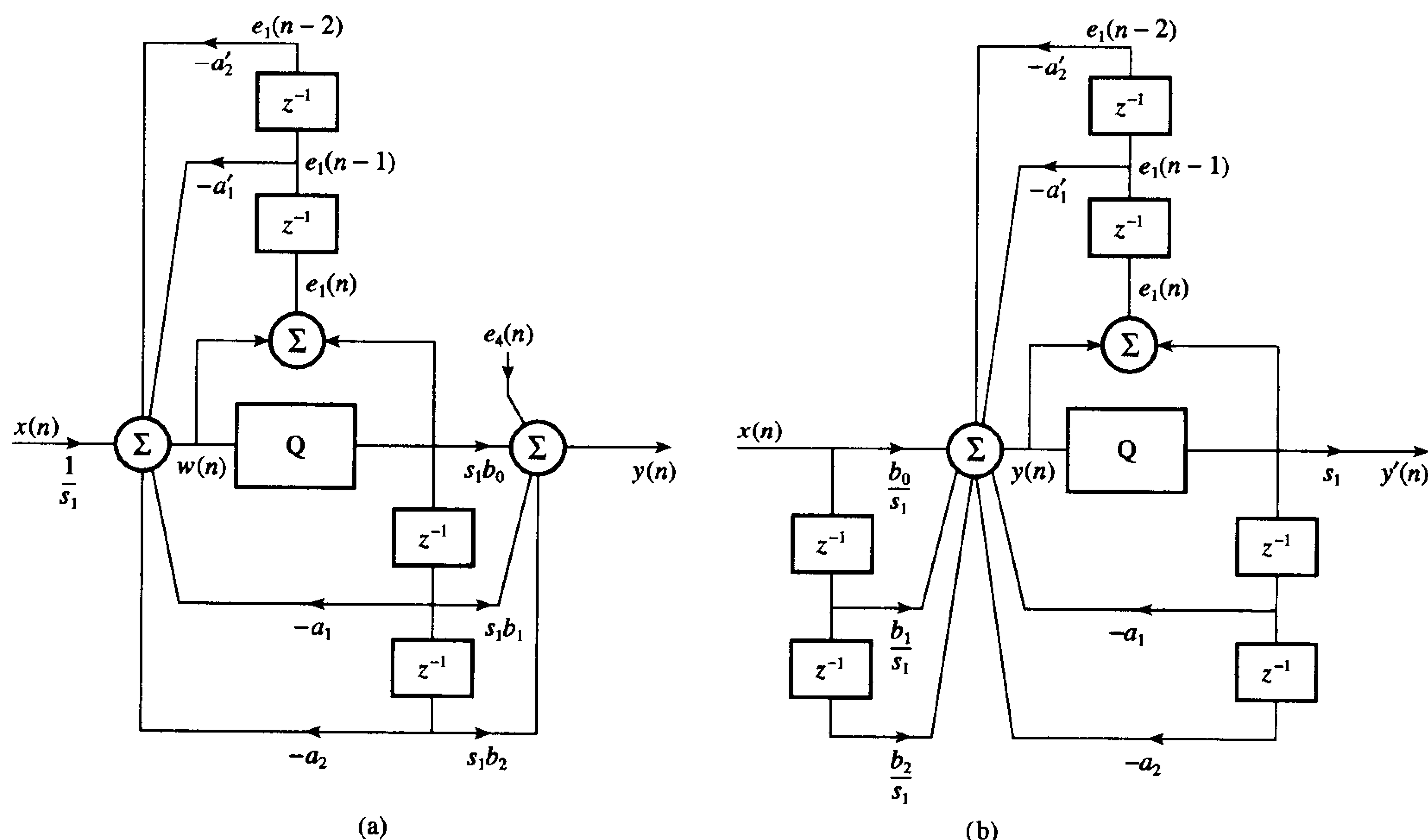


图 13.25　一个二阶 IIR 滤波器的两种不同实现结构

表 13.5　FWA 程序计算例 13.15 的结构

	噪声功率	
情景	标准型	直接型
1	$556.0108q^2$	$556.0108q^2$
2	$77.1261q^2$	$78.6247q^2$
3	$710.0842q^2$	$713.0933q^2$
4	$414.1014q^2$	$413.845q^2$
5	$12.2659q^2$	$14.9999q^2$

13.4.14　确定误差反馈系数的实际值

根据目前的讨论，显然误差频谱 $E(z)$ 受滤波器极点的影响。基本上，误差频谱被滤波器极点所放大。如果我们假定误差具有一个平坦的频谱，则滤波器输出的噪声在极点频率附近被放大。误差反馈系数通过在噪声通道上引入一个或多个零点来抵消误差频谱的放大效应。对于一个一阶误差反馈网络，在误差传递函数的分子中引入了一个单零点。对于一个二阶误差反馈网络，反馈系数在噪声传递函数中引入了两个零点。这两种情况下，零点都不会影响滤波器的输入。

一种简单但有效的策略是将反馈零点尽可能近地放在极点频率上，以抵消极点的放大效应，请参见图 13.21。在实践中，影响误差反馈系数选择的因素包括希望避免更多的舍入噪声、双精度的

使用、希望简化乘法器和需要将误差反馈网络零点尽可能近地放在滤波器极点上，以抵消它对舍入噪声的影响。由于这些原因，误差反馈系数的值通常被限于简单的整数，k_1和k_2分别为0、±1、±2。

对于反馈系数是简单整数的情况，反馈网络中的零点应放置在0、±60°、±90°、±120°、±180°处，取决于k_1和k_2的取值。误差反馈系数的可能值和相应零点的位置总结在表13.6中。

误差反馈系数的选择依赖于滤波器的类型。例如，低通滤波器在直流或者接近直流处有极点。因此，由表我们看到，k_1和k_2的选择被限制为选表中的1、3、4或5，因为只有这几种情况产生靠近滤波器极点的零。另一方面，高通滤波器有极点靠近有效频谱高频端（即在$F_s/2$附近），k_1和k_2的可能选择是1、6、7或8。

表13.6 整数误差反馈系数和相应的零点位置

序号	k_1值	k_2值	零点位置
1	0	1	一对零点在0°和180°（即在直流和$F_s/2$处）
2	0	-1	一对复共轭零点在±90°（即±$F_s/4$处）
3	1	0	单零点在0°（即直流成分上）
4	1	-1	一对复共轭零点在±60°（即±$F_s/6$处）
5	2	-1	双零点在0°（即直流成分上）
6	-2	-1	双零点在180°（即$F_s/2$处）
7	-1	-1	一对复共轭零点在±120°（即±$F_s/3$处）
8	-1	0	单零点在180°（即$F_s/2$处）

例13.16

(a) 利用适当的框图帮助，讨论定点数字IIR滤波器中的舍入噪声问题。你的回答应包含以下要点：

(i) 舍入噪声是怎样在IIR滤波器中产生的；

(ii) 舍入噪声对IIR滤波器性能的影响。

(b) 图13.26给出了一个二阶子滤波器的结构，并附带误差反馈方案。假定子滤波器使用2的补码、定点算术，量化发生在乘积的加法之后。

(i) 在输入变换$X(z)$和量化误差$E(z)$的基础上推导出量化输出的变换表达式$\hat{Y}(z)$，由此证明误差反馈网络对输入信号没有不好的影响。

(ii) 推导误差反馈函数的表达式。

(iii) 在实践中，影响误差反馈系数值的选择的主要因素是什么？

(c) 根据对极点和零点位置的分析，获得合适的整数作为误差反馈系数，最小化下列滤波器输出的舍入噪声基底：

(i) $$H(z)=\frac{1+2z^{-1}+z^{-2}}{1-1.75z^{-1}+0.81z^{-2}}$$

(ii) $$H(z)=\frac{1-2z^{-1}+z^{-2}}{1+1.75z^{-1}+0.81z^{-2}}$$

(iii) $$H(z)=\frac{1-z^{-2}}{1+0.81z^{-2}}$$

假定每个滤波器都采用图13.26的实现结构，二阶多项式：

$$1+d_1z^{-1}+d_2z^{-2}$$

的根是$r\angle\theta$和$r\angle-\theta$，其中

$$r=\sqrt{d_2}\qquad \theta=\cos^{-1}\left(\frac{-d_1}{2r}\right)$$

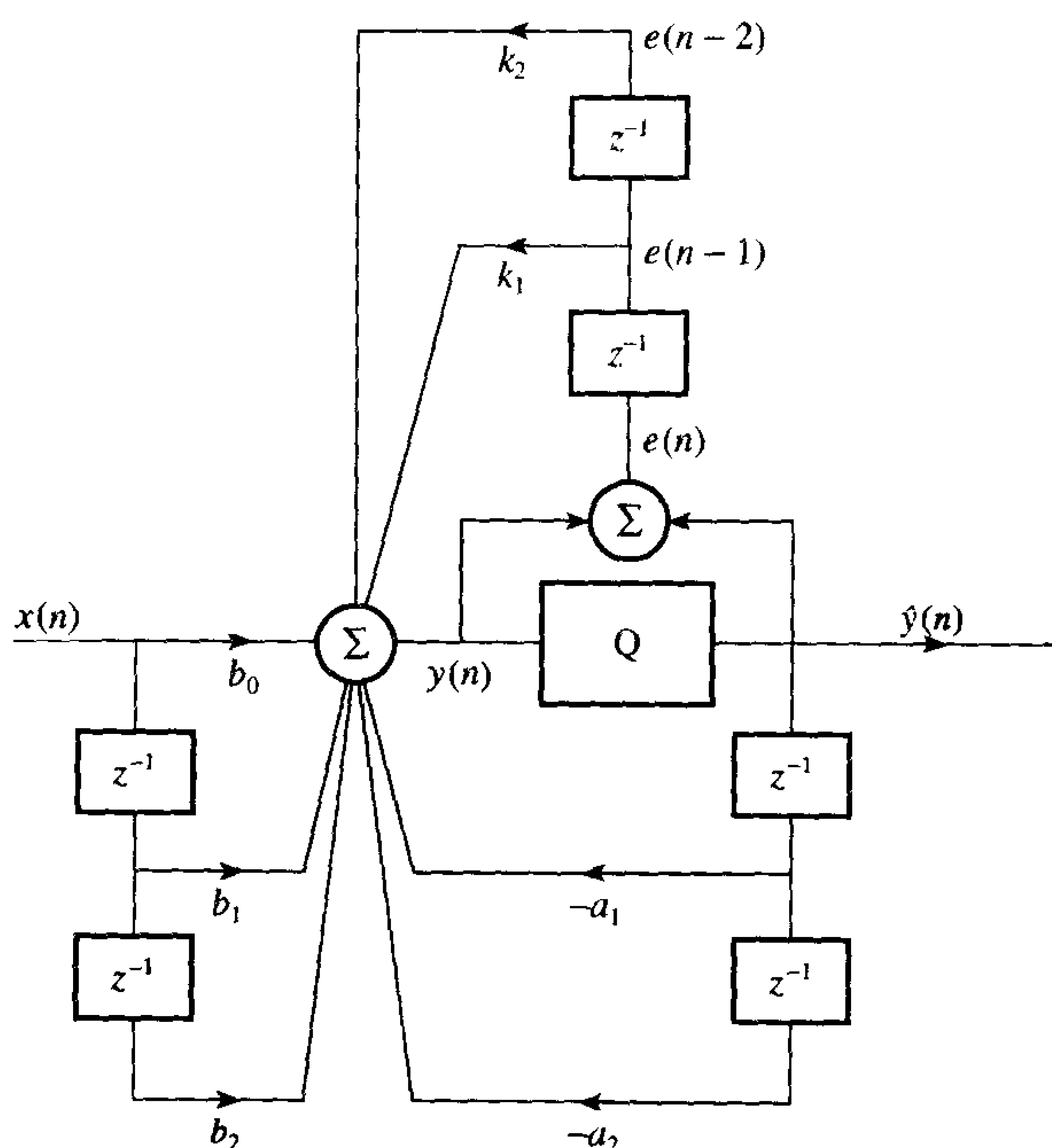


图 13.26 一个二阶噪声消减方案——误差反馈系数的选择

解:

(a) 舍入噪声是在乘积量化时产生的——舍入或截断——与/或乘积和的量化，它在递归实现中是必需的，可以使变量保持在系统的容许字长之内。例如，两个用B位表示的数的乘法，结果是一个$2B$位的数。如果结果不量化回B位，接下来结果的字长会不受限制地增长。量化误差被滤波器的极点所放大，在输出端表现为噪声。该噪声使系统整体噪声平台上升。它导致低输入水平信号的失真，在需要高保真的应用中是不能被接受的。舍入误差还会在滤波器输出端产生小幅振荡，即便是这时没有输入。滤波器的拓扑框图（例如二阶标准型子滤波器）和舍入噪声模型能够用来阐明这一回答。

(b) (i) 根据图 13.26，舍入误差 $e(n)$、量化和未量化的滤波器输出 $y'(n)$、$y(n)$之间的关系为

$$e(n) = y(n) - \hat{y}(n) \tag{13.39a}$$

$$y(n) = \sum_{i=0}^{2} b_i x(n-i) - \sum_{i=1}^{2} a_i \hat{y}(n-i) + \sum_{i=1}^{2} k_i e(n-i) \tag{13.39b}$$

利用 13.39a 式和 13.39b 式，进行 z 变换并简化得到希望的公式：

$$\hat{Y}(z) = \left(\frac{\sum_{i=0}^{2} b_i z^{-i}}{1 + \sum_{i=1}^{2} a_i z^{-i}} \right) X(z) - \left(\frac{1 - \sum_{i=1}^{2} k_i z^{-i}}{1 + \sum_{i=1}^{2} a_i z^{-i}} \right) E(z) \tag{13.39c}$$

(ii) 噪声传递函数可以通过将输入设定为零，再从上面的 13.39c 式获得：

$$H_e(z) = \frac{1 - \sum_{i=1}^{2} k_i z^{-i}}{1 + \sum_{i=1}^{2} a_i z^{-i}} = \frac{1 - k_1 z^{-1} - k_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

(iii) 考虑的主要因素是需要避免更多的舍入误差或使用双精度，需要避免使用另一个乘法器，需要是一个最小相位系统，需要将误差反馈网络的零点尽可能放在滤波器极点附近，以抵消它对舍入噪声的影响。由于这些原因，误差反馈系数的值仅限于简单的整数值，k_1和k_2分别等于0、±1、±2。

(c) (i) 利用多项式根的公式，我们发现滤波器的共轭极点在z平面的矢径$r=\sqrt{0.81}=0.9$，相角$\theta=\pm\cos^{-1}(1.75/2\times 0.9)=\pm 13.5°$；一对零点在矢径$r=1$和相角$\theta=180°$（即$F_s/2$处）。因此，该滤波器是一个低通滤波器。误差反馈系数应包含双零点在$r=1$和$\theta=0$处，以抵消极点对舍入噪声的影响，因此应该取值：$k_1=2$和$k_2=-1$。我们也可以使用一阶误差反馈系数，取值$k_1=1$和$k_2=0$（直流处的一个零点）。

(ii) 对于第二个滤波器，极点的矢径和相角分别是$r=0.9$，$\theta=\pm\cos^{-1}(-1.75/2\times 0.9)=\pm 166.4°$，一对零点在直流分量上，所以该滤波器显然是一个高通滤波器。与滤波器极点最近的反馈网络零点对应着反馈系数值$k_1=-2$和$k_2=-1$（即在$r=1$和相角$\theta=180°$处一对双零点）。

(iii) 对于第三个滤波器，有两个复共轭极点位于z平面的矢径$r=0.9$和相角$\theta=\pm 90°$处，一对零点在0°和180°处。为了尽可能地消除极点的影响，整数反馈系数的最佳选择是$k_1=0$和$k_2=-1$。如表13.6所示，这会产生一对复共轭零点在矢径1和相角±90°处。

13.4.15 乘积舍入误差产生的极限环

除了SNR的下降，舍入误差还能使滤波器输出产生振荡，或者输出停留在一个固定的非零值上，即便这时没有输入。这种效应称为低电平极限环，我们用一个例子来说明。

例13.7 一个一阶IIR滤波器，其特性为下面的差分方程：

$$y(n)=x(n)+\alpha y(n-1) \quad n>0$$

在初始条件$y(0)=6$和零输入，即$x(n)=0$（$n=0,1,\ldots$）

(1) 假定无限精度，计算和绘出前10个输出值，且(i)$\alpha=-0.75$和(ii) $\alpha=0.75$；

(2) 重复(1)，但假定数据和寄存器的长度都为4位长（即3个数据位和1个符号位），乘积被量化；

(3) 重复(1)和(2)，但假定乘法后乘积立刻被截断。

解：

三种情况下的输出抽样值在图13.27和表13.7中给出。

可以看出，如果输入$x(n)$是零，输出$y(n)$是不确定的。对于无限精度，输出呈指数形式下降到零，与α的符号无关。然而，如果使用有限精度运算，由于输出被舍入到最近的整数，则对于正的α，输出保持在一个固定的电平。输出电平的受限范围称为死区（deadband）。在本例中，死区的范围是[-2, 2]。对于一阶滤波器死区的范围是（Jackson, 1986）

$$k=\text{int}\left[\frac{0.5}{1-\|\alpha\|}\right]$$

其中int[·]代表方括号中数量的整数部分。如果α是个负数，输出在一个$F_s/2$的频率上振荡，幅值固定，只是符号改变。这是因为在没有输入时，滤波器输出下降到低于量化电平，就又被舍入到下一个电平。该过程的不断重复就产生了低电平振荡，这些低电平振荡在某些应用中是不希望看到的。例如，它们会给当说话者静默时处于空闲信道条件的电话系统产生不愉快的噪

声。一种消除极限环的方法是增加处理器字长，或在舍入前给输出加一个颤音信号。前面已讨论过的 ESS，也能降低电平极限环的幅度，有时甚至能完全消除它。

通常情况下，如果系数是在稳定三角形的区域内，舍入极限环不会存在于一个二阶滤波器中。

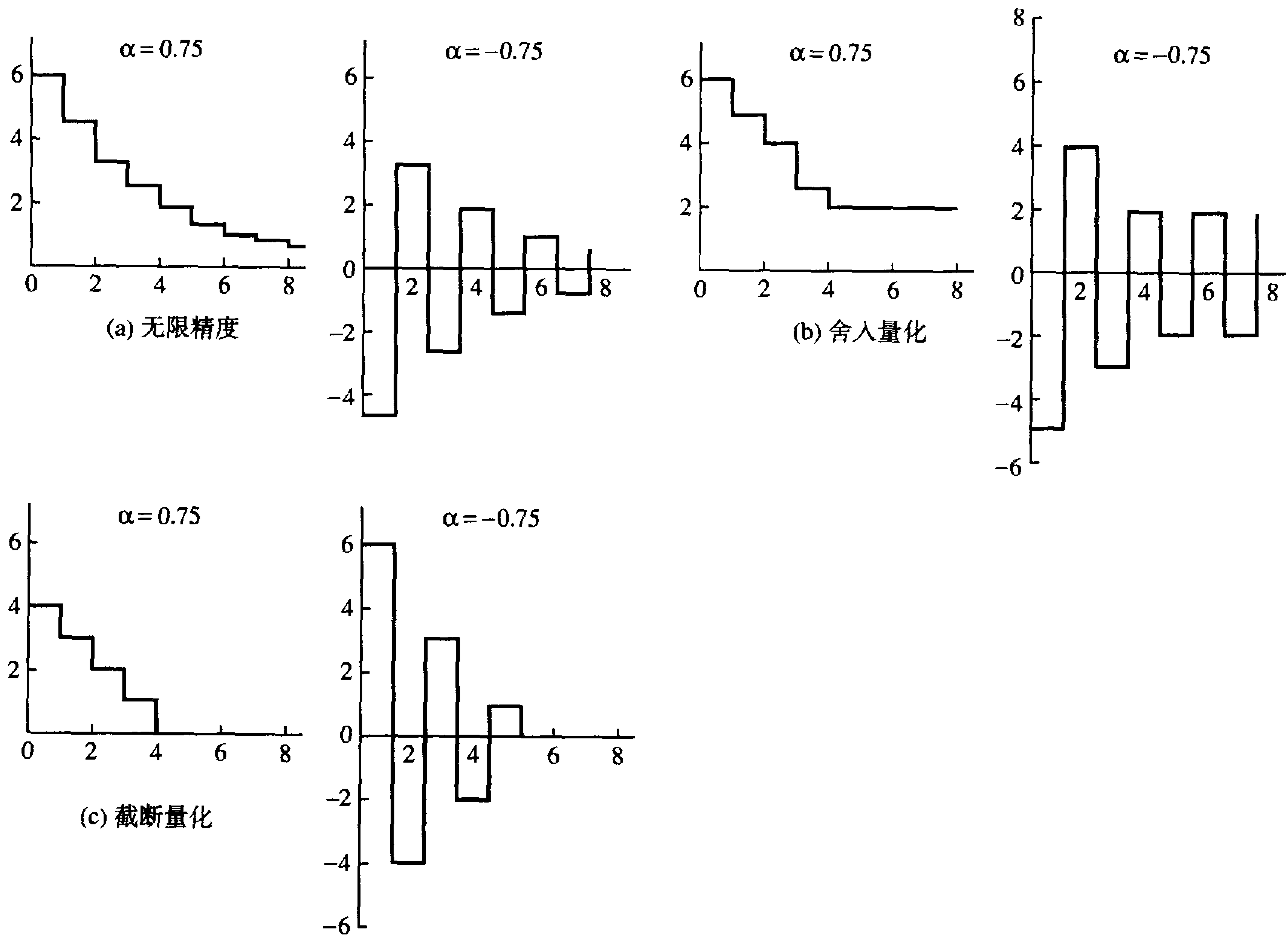


图 13.27　在一个一阶 IIR 滤波器中因乘积量化而产生低电平极限环的示意图

表 13.7　例 13.17 的结果

	$y(n)$, ($\alpha = 0.75$)			$y(n)$, ($\alpha = -0.75$)		
n	无限精度	舍入	截断	无限精度	舍入	截断
0	6	6	6	6	6	6
1	4.5	5	4	−4.5	−5	−4
2	3.38	4	3	3.38	4	3
3	2.53	3	2	−2.53	−3	−2
4	1.90	2	1	1.90	2	1
5	1.42	2	0	−1.42	−2	0
6	1.07	2	0	1.07	2	0
7	0.80	2	0	−0.80	−2	0
8	0.60	2	0	0.60	2	0
9	0.45	2	0	−0.45	−2	0

13.4.16　其他非线性现象

除了溢出和乘积极限环，其他会影响 IIR 数字滤波器性能的非线性效应有

(1) **跃迁现象（jump phenomenon）**　当给滤波器输入一个正弦波时，对同样的输入信号可能存在两个输出电平。输入信号幅度或频率的微小变化可能导致输出从一个电平到另一个。

已经证明这种现象可能存在于稳定三角形内的某些区域。在这些区域，滤波器系数满足条件 $|a_1|a_2 < -1$。ESS 已被发现能够降低这种非线性效应的结果。

(2) **子谐波响应（subharmonic response）** 对于一个正弦波输入，输出可能包含输入的子谐波（Claasen, 1974）。因此对于同样的输入信号但不同的初始条件，我们会获得差异很大的输出。这些效应在极点接近单位圆的滤波器中更加严重。

13.5 FFT 算法中的有限字长效应

在大多数 DSP 算法中，在使用定点运算来执行 FFT 算法中产生的主要误差有

- 舍入误差，当乘积 W^kB 被截断或舍入到系统字长时产生；
- 溢出误差，当蝶形运算的输出超出了容许的字长时产生；
- 系数量化误差，当使用有限位数来表示旋转因子（twiddle factor）时产生。

我们将考察基 -2 的 FFT 输出中的上述误差效应。

13.5.1 FFT 中的舍入噪声

在任何 FFT 计算中的基本操作是蝶形运算，对于基 -2 DIT 的 FFT 来说，

$$A' = A + W^kB$$
$$B' = A - W^kB$$

其中 A 和 B 是蝶形运算的输入，A' 和 B' 是输出。通常情况下，旋转因子 W^k 以及输入和输出都是复数值。在定点实现中，蝶形运算用实数算术来实现，因此我们需要用矩形形式来表示 A' 和 B'（参见第 12 章）：

$$\begin{aligned} A' &= A_r + B_r\cos(X) + B_i\sin(X) + j[A_i + B_i\cos(X) - B_r\sin(X)] \\ &= A_r + B_rW_r + B_iW_i + j[A_i + B_iW_r - B_rW_i] \end{aligned} \tag{13.40a}$$

$$\begin{aligned} B' &= A_r - [B_r\cos(X) + B_i\sin(X)] + j[A_i - \{B_i\cos(X) - B_r\sin(X)\}] \\ &= A_r - (B_rW_r + B_iW_i)_i + j[A_i - (B_iW_r - B_rW_i)] \end{aligned} \tag{13.40b}$$

其中下标 r 代表变量的实部，而下标 i 则代表虚部，$X = 2\pi k/N$。因此蝶形运算需要四个乘法和六个实数加法（我们将减法与加法同等看待）。在一个定点实现中，上面的每个乘积近似需要操作数本身位数的两倍来表示。例如，如果变量 B_r、B_i、W_r 和 W_i 每个都用 16 位数来表示，则在乘法后每个乘积都需要 32 位来表示。截断或舍入每个乘积回到 16 位都会产生一个误差，即我们熟悉的舍入误差。

因此，对于每个蝶形运算可以得到四个舍入噪声源，每个乘积产生一个，所以每个蝶形运算输出的舍入噪声功率（方差）为

$$\sigma_B^2 = 4 \times \frac{q^2}{12} \tag{13.41}$$

其中 $q = 2^{-(B-1)}$，系统字长为 B 位。

某一阶段的一个蝶形运算中产生的噪声会进入下一阶段。如果我们查看一个 FFT 的流图，如图 13.28 所示，我们发现每个 FFT 的输出 $X(k)$，可以回溯到$(N-1)$个蝶形。图 13.29 显示了对输出 $X(2)$做出贡献的蝶形，其中 $N = 7$。一般来说，每个 FFT 输出与 1 级的 $N/2$ 个蝶形有关，与 2 级的

$N/4$ 个蝶形有关，与 3 级的 $N/8$ 个蝶形有关，依次类推。假定每个蝶形都产生了相同但不相关的误差，则每个 FFT 输出 $X(k)$ 的最大噪声功率为（Oppenheim and Weinstein, 1972）

$$\sigma_0^2 = (N-1)\sigma_B^2 \approx N\sigma_B^2 = \frac{N}{3}2^{-2(B-1)} \text{（当} N \text{较大时）}$$

因此，噪声功率与变换的大小直接成正比。将 N 加倍，即等效于给 FFT 加一级，同样使噪声功率加倍。为了保持相同的噪声功率，我们可以增加字长 1 位，因为噪声功率正比于 N 和 $2^{-2(B-1)}$。

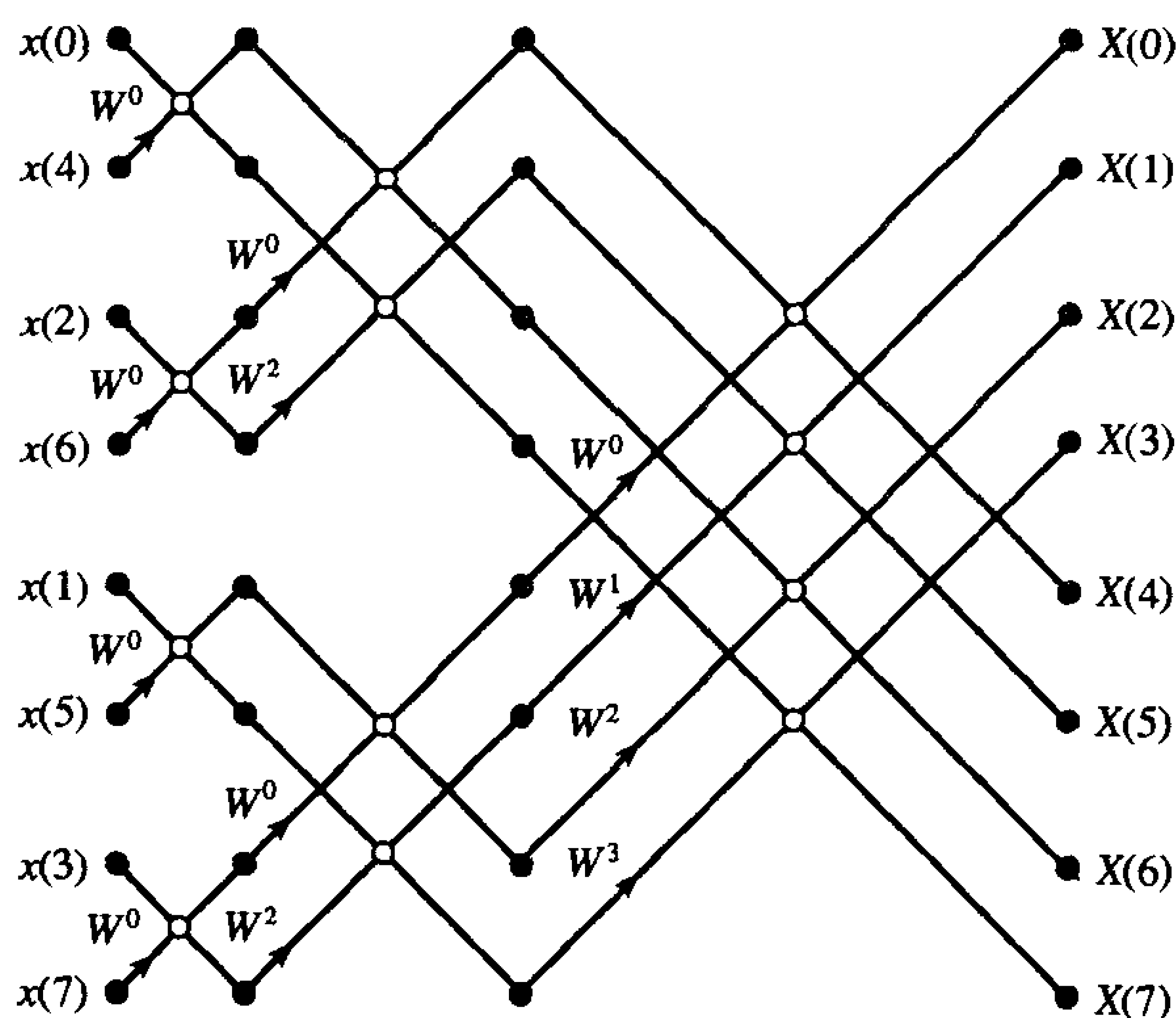


图 13.28 一个 8 点、基-2、时间抽取的 FFT 算法流图

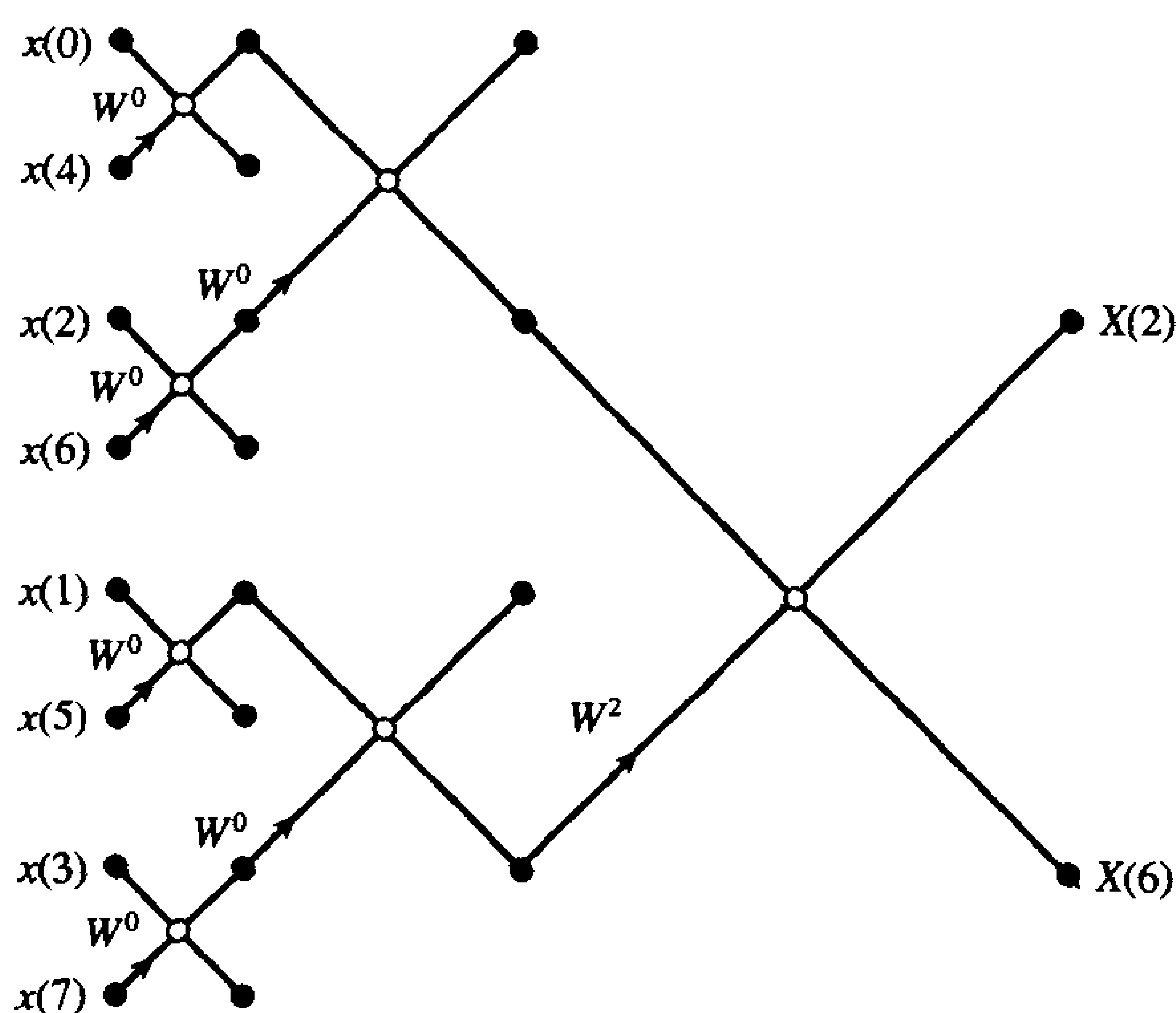


图 13.29 用以显示对输出 $X(2)$ 和 $X(6)$ 的舍入噪声做出贡献的蝶形的流图

这种情况下的信噪比近似等于

$$SNR = \frac{1/3}{N2^{-2(B-1)}/3} = \frac{2^{2(B-1)}}{N}$$

如果我们只考虑没有微小旋转因子的蝶形的噪声贡献（蝶形的旋转因子 $W^k = \pm 1$ 或 $\pm j$，则会得到准确无误的乘积），则舍入误差产生的噪声功率很小。实际上，如果利用这一信息，我们会发现有些 FFT 的输出没有误差产生。所以，上面的表达式代表了 SNR 的上限。

13.5.2 溢出误差和 FFT 中的伸缩变换

在FFT运算中进行伸缩变换对于避免溢出是必要的（在执行13.40a式和13.40b式的加法之后），因为每次蝶形运算后的数据尺寸在增大。在FFT运算中有许多方法可以对数据进行伸缩变换以避免溢出。一种通用的伸缩变换方案是基于观察每个蝶形的输出是否满足关系（Oppenheim and Weinstein, 1972）：

$$\max[|A'|,|B'|] \leqslant 2\max[|A|,|B|] \tag{13.42}$$

这暗示着级数每增加1级，蝶形输出的最大模数乘以因子2。这样，如果每个蝶形的输入都按0.5进行伸缩变换，则在输入数据的幅度在容许字长范围之内的情况下，输出将不应发生溢出，请参见图13.30。

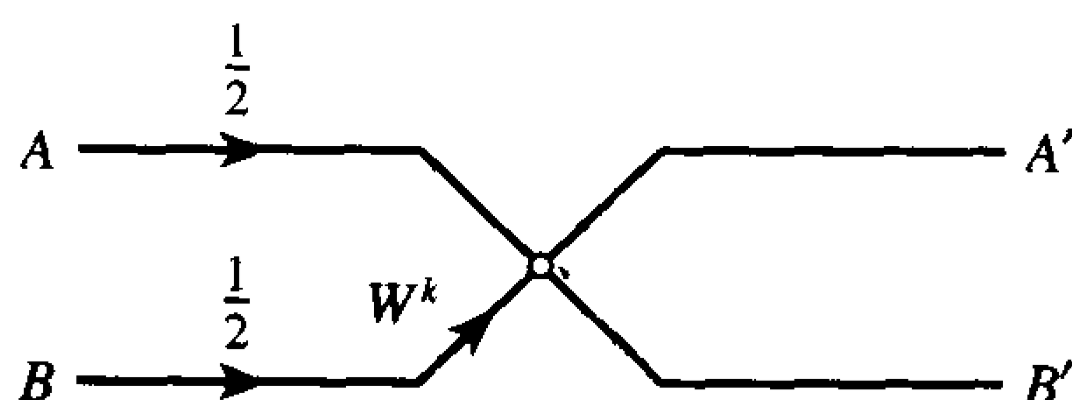


图 13.30 一种降低每个蝶形中的溢出的伸缩变换方案

实际上，某些条件下即便输入进行0.5的伸缩变换且输入小于1，也不足以避免溢出。为了说明这个问题，考虑13.40式的显式表达：

$$A' = A_r + B_r\cos(X) + B_i\sin(X) + j[A_i + B_i\cos(X) - B_r\sin(X)]$$

$$B' = A_r - [B_r\cos(X) + B_i\sin(X)] + j[A_i - \{B_i\cos(X) - B_r\sin(X)\}]$$

如果 $X = 2\pi k/N = 45°$，则 $\cos(45°) = \sin(45°) = \sqrt{2}/2$。在不进行伸缩变换和输入的实部和虚部均设为1（有限条件下）时，我们从上面公式得到

$$A' = 2.4142 + j;\quad B' = -0.4142 + j$$

当每个输入进行0.5的伸缩变换后，则有

$$A' = 1.2071 + 0.5j;\quad B' = -0.2071 + 0.5j$$

显然，在进行0.5的伸缩变换的情况下，A' 的实部仍会产生一个溢出，因为它的幅度超过了1。

尽管存在溢出的可能，由于做法简便——一个简单的向右位移（或者在定点乘法之后，利用其通常产生的两个符号位，我们不用进行任何操作），大多数实现仍只采用一个0.5的伸缩因子。为了在所有情况下都避免溢出，输入应该先按1.2071（2.414 21/2）进行伸缩变换，然后每级按因子0.5进行伸缩变换。在FFT之后，输出被增幅回到正确的值。对于大多数真实数据，对输入进行额外的伸缩变换是不必要的，因为最大值不可能达到。

对蝶形输入进行伸缩变换改变了FFT的舍入噪声特性。这时的输出信噪比近似为

$$SNR = \frac{1}{2N}2^{2(B-1)} \tag{13.43}$$

例 13.18 一个硬件FFT处理器在它的蝶形运算中使用定点算术。估计执行一个1024点FFT且输出SNR为40 dB所需的最大字长。假定在整个FFT中，每个蝶形的输入按0.5进行伸缩变换。

解：

$$40 = 10\log\left(\frac{1}{2N}2^{2(B-1)}\right);\quad 10^{\frac{40}{10}} = \frac{1}{2N}2^{2(B-1)}$$

$$B - 1 = \tfrac{1}{2}\log_2(2N\times 10^4)/\log(2) = 12.14 = 13 \text{（近似的）}$$

系统字长 $B = 14$ 位。

13.5.3 FFT 中的系数量化

在许多硬件 FFT 实现中，旋转因子 W^k 的实部和虚部通常预先计算好，然后量化成 B 位存储在一个查找表中，这里 B 位是系统字长。这就产生了熟悉的量化误差。

13.6 小结

一个DSP系统的性能受限于其实现所使用的位数。四种通常的误差源是(1)输入量化、(2)系数量化、(3)乘积舍入和(4)加法溢出。本章分析它们对DSP系统性能的影响，进一步给出了消除或最小化的技术。我们将 IIR 滤波器作为主要的分析对象。系数字长必须足够以最小化系数量化对频率响应的影响，还要预防可能出现的不稳定。一个 IIR 滤波器的稳定性永远是需要首先考虑的。在无限精度下稳定的一个 IIR 滤波器可能会在有限精度下变得不稳定。因此在高保真音响中，一般需要24位系数才能处理低频音频信号。在其他情况下，使用 16 或更多的位数表示系数，并且使用双精度加法器实现算术操作就足以最小化有限字长的影响。

截断或舍入误差源来自于有限精度的算术操作对滤波器产生了一个非线性效应（比如极限环），当没有输入或输入为常数时滤波器的输出产生振荡。舍入误差对滤波器性能的影响可以量化成滤波器输出的 SNR。由于舍入误差而导致的 SNR 降低可以通过误差频谱整形（ESS）方案来弥补。这类方案的主要作用是消除滤波器极点对舍入误差的“放大”效应。其代价是乘法和加法运算次数的增加，不过一阶整系数 ESS 的计算是有效的。

设计程序在指导手册的CD上提供，可以使设计者计算滤波器系数和分析一些有限字长对滤波器性能的影响（参见前言）。

习题

13.1　图 13.31 显示了一个标准二阶子滤波器。

(1) 解释为什么在节点 1 和节点 3 的溢出是允许的，而在节点 2 则不行。

(2) 寻找合适的伸缩因子来降低节点 2 溢出的可能性。

(3) 假定滤波器采用一个 8 位系统实现，在加法前进行乘积量化。确定至少降低舍入噪声 20 dB 所需的额外位数。

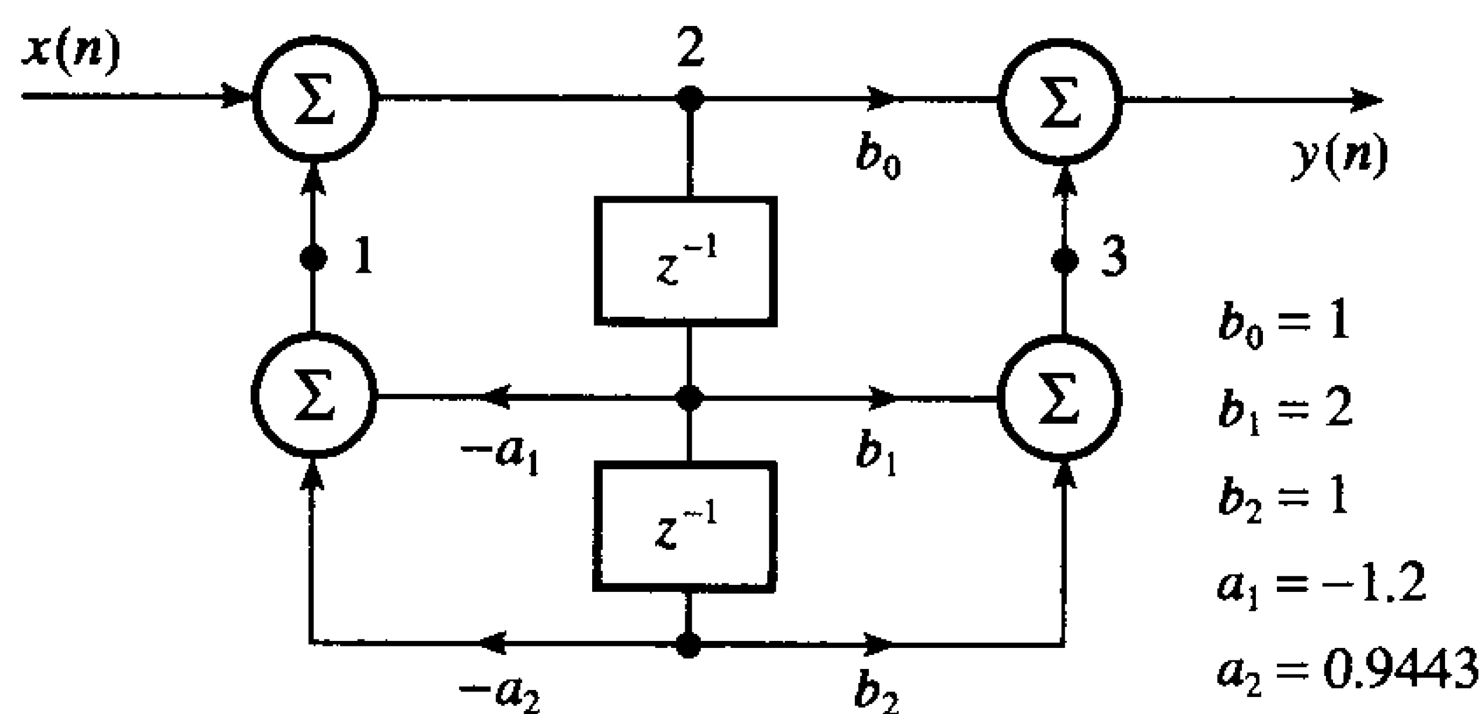


图 13.31　习题 13.1 的标准二阶子滤波器

13.2　一个 IIR 滤波器具有如下传递函数：

$$H(z) = \frac{0.1436 + 0.2872z^{-1} + 0.1436z^{-2}}{1 - 1.8353z^{-1} + 0.9748z^{-2}}$$

(1) 确定极点和零点的位置，绘出极零图。

(2) 确定极点到原点的径向距离。

(3) 估计表示每个系数所需要的位数，从而实现

(a) 保持稳定；

(b) 通带幅频响应的失真度小于 1%。

13.3 下面是滤波器的传递函数：

$$H(z)=\frac{1-0.9631z^{-1}+z^{-2}}{1-1.5763z^{-1}+0.9413z^{-2}}$$

(1) 寻找合适的伸缩因子，在采用标准型二阶子滤波器实现它时避免产生溢出。

(2) 为得到 60 dB 的输出信噪比，确定所需的最小字长。说明所用的假设。

13.4 一个八阶 IIR 滤波器的极点和零点如下：

极点	零点
0.2870 ± 0.9075j	0.0553 ± 0.9985j
0.7882 ± 0.5658j	0.8828 ± 0.4698j
0.4089 ± 0.7447j	−0.4816 ± 0.8764j
0.6479 ± 0.5975j	0.9617 ± 0.2740j

(1) 绘出极零图，配对极点和零点，调整你的配对方案。

(2) 根据极零图写出滤波器的传递函数。假定滤波器用串联实现，确定一个合适的排序方案。

(3) 应用指导手册的 CD 中的有限字长分析程序，确定子滤波器的合适的伸缩因子。

(4) 假定输入数据被数字化成 8 位，由于舍入误差而导致的 SNR 下降不超过 0.5 dB。确定内部数据、系数和数据变量的合适字长。

13.5 传递函数如下：

$$H(z)=\frac{1-1.4890z^{-1}+z^{-2}}{1-0.3724z^{-1}+0.5119z^{-2}}\times\frac{1-1.9020z^{-1}+z^{-2}}{1-0.3779z^{-1}+0.0851z^{-2}}$$

(1) 确定和绘出极零点的位置。

(2) 当采用 48 kHz 的抽样频率时，绘出滤波器的幅频和相频响应。

(3) 写出 2 的补码的 8 位（包括符号位）定点数表示的滤波器系数。

(4) 对量化后的滤波器重复(1)和(2)，比较两组结果。

13.6 传递函数如下：

$$H(z)=0.1436\frac{1+2z^{-1}+z^{-2}}{1-0.67993z^{-1}+0.49133z^{-2}}$$

(1) 确定合适的伸缩因子以避免加法器 1 的输出溢出，以及输出伸缩因子以使总增益为 1。

(2) 使用 8 位定点算术将滤波器系数和伸缩因子编码。

(3) 确定总的舍入噪声。

13.7 一个低通 IIR 滤波器用于在数字电话中带限语音信号。滤波器需要满足下列特性：

通带	0~3300 Hz
阻带	4.6~16 kHz
通带波纹	< 0.1 dB

阻带衰减	> 30 dB
ADC	12 位
系数字长	16 位

假定抽样频率为 32 kHz，确定

(1) 合适的传递函数，假定滤波器采用二阶与 / 或一阶子滤波器的串联实现；

(2) 每个子滤波器的伸缩因子；

(3) 由于系数量化而导致的通带和阻带波纹变化；

(4) 由于舍入噪声而导致的 SNR 下降，假定应用加法后量化。

13.8 在 IIR 滤波器中应用伸缩变换以避免加法器溢出。一种方案是将每个 IIR 子滤波器的输入衰减，其伸缩因子为

$$s_1^2 = \frac{1}{2\pi \mathrm{j}} \oint \frac{z^{-1}\mathrm{d}z}{D(z)D(z^{-1})}$$

其中 $\oint$ 代表沿单位圆的积分，即 $|z| = 1$，且

$$D(z) = 1 + a_1 z^{-1} + a_2 z^{-2}$$

(1) 找到 s_1^2 的通用表示。

(2) 计算伸缩因子 s_1，滤波器具有下面的传递函数：

$$H(z) = \frac{1 + 1.2173z^{-1} + z^{-2}}{1 + 0.9140z^{-1} + 0.8793z^{-2}}$$

13.9 设计一个切比雪夫低通 IIR 数字滤波器，满足下面特性：

通带边频	12 kHz
阻带边频	16 kHz
通带波纹	0.5 dB
阻带衰减	60 dB
抽样频率	48 kHz

假定滤波器在一个基于 TMS320C54 的系统和 12 位的 ADC 和 DAC 上实现。

13.10 设计一个切比雪夫高通 IIR 数字滤波器，满足下面特性：

通带边频	12 kHz
阻带边频	8 kHz
通带波纹	0.5 dB
阻带衰减	60 dB
抽样频率	48 kHz

假定滤波器在一个基于 DSP56300 的系统和 16 位的 ADC 和 DAC 上实现。

13.11 需要一个数字陷波（notch）滤波器以降低主要干扰源的影响。滤波器需要满足下列特性：

陷波频率	50 Hz
陷波宽度（3 dB）	±2 Hz
抽样频率	500 Hz
滤波器阶数	2

(a) 根据极 – 零点配置的方法，确定一个合适的陷波数字滤波器的传递函数。利用传递函数的帮助，解释为什么滤波器的幅频响应总体是平坦的，除了在陷波频率上。

(b) 基于频率响应寻找(a)的合适的伸缩因子，以降低内部溢出的可能性。假定滤波器用二阶标准型子滤波器来实现。

(c) 如果系数被量化成8位，确定陷波频率的改变。

13.12 (a) 利用合适的框图帮助，讨论定点数字 IIR 滤波器中的舍入噪声问题。你的回答应包括下面几点：

- IIR 滤波器中的舍入噪声是如何产生的；
- 舍入噪声对 IIR 滤波器性能的影响。

(b) 图13.32给出了一个二阶子滤波器的结构，包括一个误差反馈方案。假定子滤波器使用2的补码、定点算术实现，其量化在乘积的加法之后进行。对于输入变换$X(z)$和量化误差$E(z)$，推导量化输出的变换$\hat{Y}(z)$，由此证明误差反馈网络对输入信号没有不好的影响。

(i) 推导误差反馈函数的表达式。

(ii) 利用合适的频率响应图，解释误差反馈网络对滤波器输出的舍入噪声的影响。

(iii) 在实践中，影响误差反馈系数值的选择的主要因素是什么？

(iv) 根据对极点和零点位置的分析，获得合适的整数对来作为误差反馈系数，最小化下列滤波器输出的舍入噪声基底：

(1) $H(z)=\dfrac{1+2z^{-1}+z^{-2}}{1-1.25z^{-1}+0.81z^{-2}}$

(2) $H(z)=\dfrac{1+2z^{-1}+z^{-2}}{1+1.40z^{-1}+0.53z^{-2}}$

(3) $H(z)=\dfrac{1-2z^{-1}+z^{-2}}{1-1.4z^{-1}+0.53z^{-2}}$

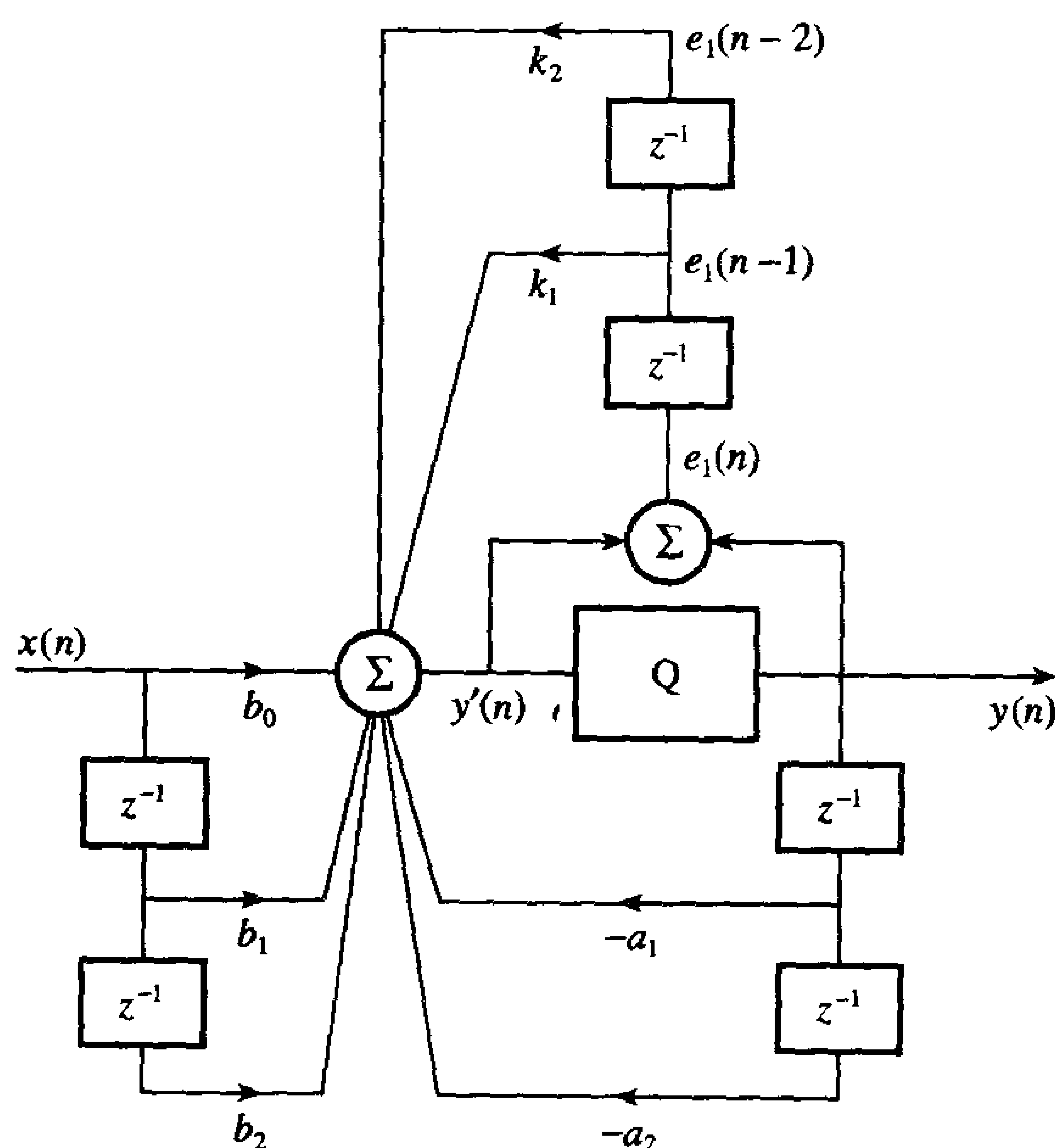


图 13.32　习题 13.12 的二阶舍入噪声消减方案

13.13 图 13.33 显示了一个简单的一阶 IIR 滤波器，并带有一个误差频谱整形方案以最小化滤波器输出端的乘积舍入噪声。分析和确定

(1) 一个合适的 L_2 伸缩因子以降低溢出的可能性；

(2) 在下面情况中由于舍入误差而产生的输出噪声功率：

(a) 没有误差反馈，即 $k'=0$；

(b) 误差反馈系数 $k'=1$。

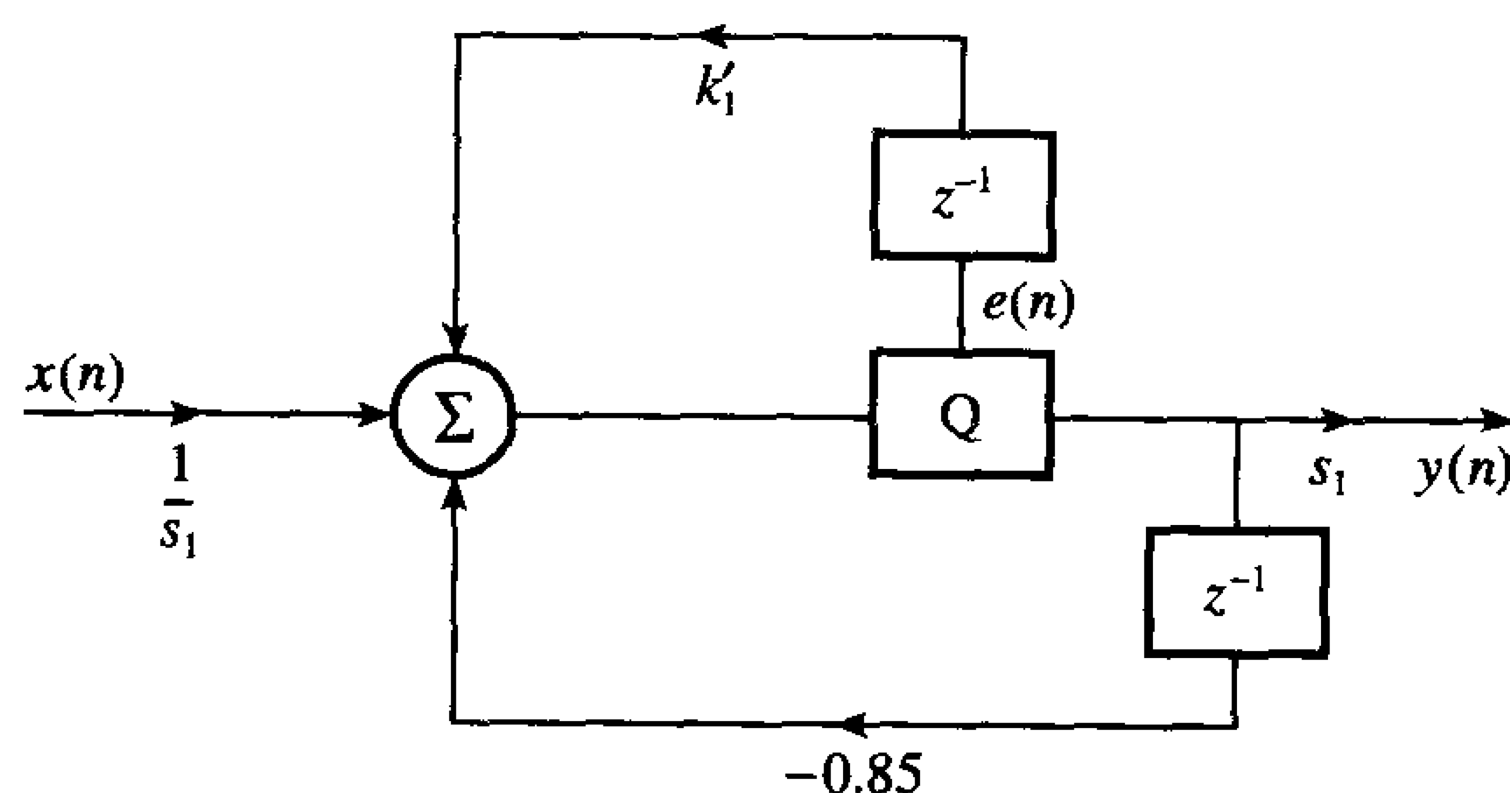


图 13.33　带误差频谱整形的一阶 IIR 滤波器

参考文献

Abu-el-Haija A. and Al-Ibrahim M.M. (1986) Improving performance of digital sinusoidal oscillators by means of error feedback circuits. *IEEE Trans. Circuits and Systems*, **33**(4), 373–80.

Antoniou A. (1979) *Digital Filters Analysis and Design*. New York: McGraw-Hill.

Chen W. (1996) Performance of cascade and parallel IIR filters. *J. Audio Eng. Soc.*, **44**(3), 148–58.

Claasen T. (1974) Improvement of overflow behaviour of 2nd-order digital filters by means of error feedback. *Electronics Lett.*, **10**(12), 240–1.

Dattorro J. (1988) The implementation of recursive digital filters for high-fidelity audio. *J. Audio Eng. Soc.*, **36**(11), 851–78.

Flores I. (1963) *The Logic of Computer Arithmetic*. Englewood Cliffs NJ: Prentice-Hall.

Higgins W.E. and Munson D. (1982) Noise reduction strategies for digital filters: error spectrum shaping versus the optimal linear state-space formulation. *IEEE Trans. Acoustics, Speech and Signal Processing*, **30**(6), 963–73.

IEEE (1979) *Programs for Digital Signal Processing*. New York: IEEE Press.

IEEE (1985) IEEE Standard for Binary Floating Point Arithmetic. *SIGPLAN Notices*, **22**(2), 9–25.

Ifeachor E.C. (2001) *A Practical Guide for MATLAB and C Language Implementation of DSP Algorithms*. Harlow: Pearson Education.

Jackson L.B. (1986) *Digital Filters and Signal Processing*. Boston MA: Kluwer.

Mitra S.K., Hirano K. and Sakaguchi H. (1974) A simple method of computing the input quantization and multiplication roundoff errors in a digital filter. *IEEE Trans. Acoustics, Speech and Signal Processing*, **22**(5), 326–9.

Mullis C.T. and Roberts R.A. (1982) An interpretation of error spectrum shaping in digital filters. *IEEE Trans. Acoustics, Speech and Signal Processing*, **30**(6), 1013–15.

Oppenheim A.V. and Weinstein C.J. (1972) Effects of finite register length in digital filtering and the fast Fourier transform. *Proc. IEEE*, **60**, 957–76.

Patterson D.A. and Hennessy J.L. (1990) *Computer Architecture: A Quantitative Approach*. San Mateo CA: Morgan Kaufmann.

Rabiner L.R. and Gold B. (1975) *Theory and Applications of Digital Signal Processing*. Englewood Cliffs NJ: Prentice-Hall.

Rader C.M. and Gold B. (1967) Effects of parameter quantization on the poles of a digital filter. *Proc. IEEE*, **55**, 688–9.

Texas Instruments (1986) *Digital Signal Processing Applications with the TMS320 Family: Theory, Algorithms and Implementations*. Texas Instruments.

Tomarakos J. and Ledger D. (1998) Using the Low-cost, High-performance ADSP-21065L Digital Signal Processor for Digital Audio Applications. Analog Devices DSP Application. Details are available at *www.analog.com*

Weitek (1984) *High Speed Digital Arithmetic VLS Application Seminar Notes*. Sunnyvale CA: Weitek.

Wilson R. (1993) Filter topologies. *J. Audio Eng. Soc.*, **41**(9), 667–78.

参考书目

Abu-el-Haija A.I. and Peterson A.M. (1979) An approach to eliminate roundoff errors in digital filters. *IEEE Trans. Acoustics, Speech and Signal Processing*, **27**, 195–8.

Ahmed N. and Natarajan T. (1983) *Discrete-time Signals and Systems*. Reston VA: Reston Publishing Inc.

Avenhaus E. (1972) Filters with coefficients of limited wordlength. *IEEE Trans. Audio Electroacoustics*, **20**, 206–12.

Barnes C.W., Tran B.N. and Leung S.H. (1985) On the statistics of fixed-point roundoff error. *IEEE Trans. Acoustics, Speech and Signal Processing*, **33**, 595–606.

Chang T.L. (1978) A low roundoff noise digital filter structure. In *Proc. IEEE Int. Symp. on Circuits and Systems*, May 1978, pp. 1004–8.

Chang T.L. (1979) Error-feedback digital filters. *Electronics Lett.* 348–9.

Chang T.L. (1980) Comments on 'An approach to eliminate roundoff errors in digital filters'. *IEEE Trans. Acoustics, Speech and Signal Processing*, **28**(2), 244–5.

Chang T.L. (1981) Suppression of limit cycles in digital filters designed with one magnitude-truncation quantizer. *IEEE Trans. Circuits and Systems*, **28**(2), 107–11.

Chang T.L. (1981) On low-roundoff noise and low-sensitivity digital filter structures. *IEEE Trans. Acoustics, Speech and Signal Processing*, **29**(5), 1077–80.

Chang T.L. and White S.A. (1981) An error cancellation digital-filter structure and its distributed-arithmetic implementation. *IEEE Trans. Circuits and Systems*, **28**(4), 339–42.

Charalambous C. and Best M.J. (1974) Optimization of recursive digital filters with finite wordlengths. *IEEE Trans. Acoustics, Speech and Signal Processing*, **22**(6), 424–31.

Claasen T.A.C.M. and Kristiansson L.O.G. (1975) Necessary and sufficient conditions for the absence of overflow phenomena in a second order recursive digital filter. *IEEE Trans. Acoustics, Speech and Signal Processing*, **23**(6), 509–15.

Claasen T.A.C.M., Mecklenbrauker W.F.G. and Peek J.B.H. (1973) Second-order digital filter with only one magnitude-truncation quantiser and having practically no limit cycles. *Electronics Lett.*, **9**, 531–2.

Claasen T.A.C.M., Mecklenbrauker W.F.G. and Peek J.B.H. (1973) Some remarks on the classification of limit cycles in digital filters. *Philips Research Rep.*, **28**, 297–305.

Claasen T., Mecklenbrauker W.F.G. and Peek J.B.H. (1975) Frequency domain criteria for the absence of zero-input limit cycles in nonlinear discrete-time systems, with applications to digital filters. *IEEE Trans. Circuits and Systems*, **22**, 232–9.

Claasen T.A.C.M., Mecklenbrauker W.F.G. and Peek J.B.H. (1976) Effects of quantization and overflow in recursive digital filters. *IEEE Trans. Acoustics, Speech and Signal Processing*, **24**(6), 517–28.

Crochiere R.E. (1975) A new statistical approach to the coefficient wordlength problem for digital filters. *IEEE Trans. Circuits and Systems*, **22**, 190–6.

Crochiere R.E. and Oppenheim A.V. (1975) Analysis of linear digital networks. *Proc. IEEE*, **63**(4), 581–94.

Diniz P.S.R. and Antoniou A. (1985) Low-sensitivity digital filter structures which are amenable to error-spectrum shaping. *IEEE Trans. Circuits and Systems*, **32**(10), 1000–7.

Elliot D.F. (ed.) (1987) *Handbook of Digital Signal Processing*. London: Academic Press.

IEEE (1978) *Digital Signal Processing II*. Institute of Electrical and Electronics Engineers.

Jackson L.B. (1970) On the interaction of roundoff noise and dynamic range in digital filters. *BSTJ*, **49**(2), 159–84.

Jackson L.B. (1976) Roundoff noise bounds derived from coefficient sensitivities for digital filters. *IEEE Trans. Circuits and Systems*, **23**(8), 481–5.

Knowles J.B. and Olcayto E.M. (1968) Coefficient accuracy and digital filter response. *IEEE Trans. Circuit Theory*, **15**, 31–41.

Liu B. (1971) Effect of finite wordlength on the accuracy of digital filters – a review. *IEEE Trans. Circuit Theory*, **18**, 670–7.

Liu B. and Kaneko T. (1969) Error analysis of digital filters realized with floating-point arithmetic. *Proc. IEEE*, **57**(10), 1735–47.

Markel J.D. and Gray A.H. (1975) Fixed-point implementation algorithms for a class of orthogonal polynomial filter structures. *IEEE Trans. Acoustics, Speech and Signal Processing*, **23**(5), 486–94.

Markel J.D. and Gray A.H. (1975) Roundoff noise characteristics of a class of orthogonal polynomial structures. *IEEE Trans. Acoustics, Speech and Signal Processing*, **23**(5), 473–86.

Motorola (1988) *Digital Stereo 10-band Graphic Equalizer Using the DSP56001*. Motorola Application Note.

Mullis C.T. and Roberts R.A. (1976) Round-off noise in digital filters: frequency transformations and invariants. *IEEE Trans. Acoustics, Speech and Signal Processing*, **24**(6), 538–50.

Munson D.C. and Liu B. (1980) Low-noise realization for narrow-band recursive digital filters. *IEEE Trans. Acoustics, Speech and Signal Processing*, **28**, 41–54.

Nagle H.T. and Nelson V.P. (1981) Digital filter implementation on 16 bit microcomputers. *IEEE Micro*, **1**, 23–41.

Oppenheim A.V. and Schafer R.W. (1975) *Digital Signal Processing*. Englewood Cliffs NJ: Prentice-Hall.

Peled A., Liu B. and Steiglitz K. (1974) A new hardware realization of digital filters. *IEEE Trans. Acoustics, Speech and Signal Processing*, **22**, 456–62.

Peled A., Liu B. and Steiglitz K. (1975) A note on implementation of digital filters. *IEEE Trans. Acoustics, Speech and Signal Processing*, **23**, 387–9.

Rabiner L.R., Cooley J.W., Helms H.D., Jackson L.B., Kaiser L.F., Rader C.M., Schafer R.W., Steiglitz K. and Weinstein C.J. (1972) Terminology in digital signal processing. *IEEE Trans. Audio and Electroacoustics*, **20**, 322–37.
Sandberg I.W. and Kaiser J.F. (1972) A bound on limit cycles in fixed-point implementations of digital filters. *IEEE Trans. Audio and Electroacoustics*, **20**, 110–12.
Sim P.K. and Pang K.K. (1985) Effects of input-scaling on the asymptotic overflow-stability properties of second recursive digital filters. *IEEE Trans. Circuits and Systems*, **32**(10), 1008–15.
Steiglitz K. (1971) Designing short-word recursive digital. *Proc. 9th Ann. Allerton Conf. on Circuit and System Theory*, 6–8 October, pp. 778–88.
Steiglitz K., Bede L. and Liu B. (1976) An improved algorithm for ordering poles and zeros of fixed-point recursive digital filters. *IEEE Trans. Acoustics, Speech and Signal Processing*, **24**, 341–3.
Taylor F.J. (1983) *Digital Filter Design Handbook*. New York: Marcel Dekker.
Thong T. (1976) Finite wordlength effects in the ROM digital filter. *IEEE Trans. Acoustics, Speech and Signal Processing*, **24**, 436–7.
Thong T. and Liu B. (1977) Error spectrum shaping in narrowband recursive digital filters. *IEEE Trans. Acoustics, Speech and Signal Processing*, **25**, 200–3.
Williamson D. and Sridharan S. (1985) An approach to coefficient wordlength reduction in digital filters. *IEEE Trans. Circuits and Systems*, **32**(9), 893–903.

附录

13A　IIR 滤波器的有限字长分析程序

C语言编程的IIR滤波器的有限字长分析程序与演示性示例，包含在本书的指导手册（Ifeachor, 2001）中（详见前言）。

13B　L_2 伸缩因子公式

标准型子滤波器在图 13B.1 中给出。子滤波器的传递函数为

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \tag{13B.1}$$

用以降低节点 $w(n)$处溢出可能性的 L_2 伸缩因子为

$$s_1^2 = \frac{1}{2\pi j}\oint \frac{z^{-1}\,dz}{D(z)D(z^{-1})} = \frac{1}{2\pi j}\oint F(z)\,dz \tag{13B.2}$$

其中

$$D(z) = 1 + a_1 z^{-1} + a_2 z^{-2}$$
$$F(z) = z^{-1}/D(z)D(z^{-1})$$

$\oint$ 代表围绕圆 $|z| = 1$ 的围线积分。

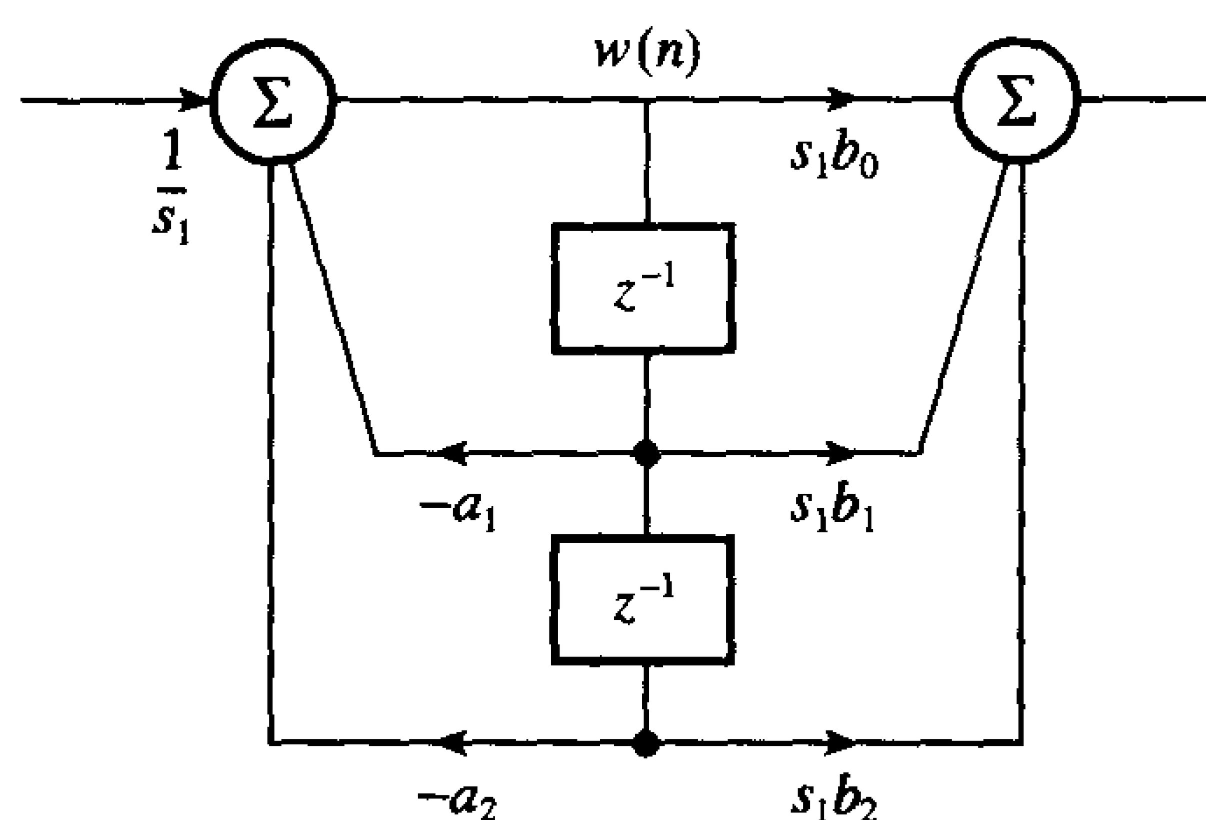


图 13B.1 标准型子滤波器

利用 13B.2 式中的 $D(z)$值，我们有

$$s_1^2=\frac{1}{2\pi\mathrm{j}}\oint\frac{z^{-1}\,\mathrm{d}z}{(1+a_1z^{-1}+a_2z^{-2})(1+a_1z+a_2z^2)}$$

$$=\frac{1}{2\pi\mathrm{j}}\oint\frac{z\,\mathrm{d}z}{(z^2+a_1z+a_2)(1+a_1z+a_2z^2)}$$

单位圆积分域内的极点 z_1 和 z_2 可由下式求得

$$z^2+a_1z+a_2=(z-z_1)(z-z_2)=0 \tag{13B.3}$$

通过计算留数，s_1^2 是$F(z)$的留数的和:

$$\begin{aligned}
s_1^2&=\lim_{z\to z_1}\frac{(z-z_1)z}{(z^2+a_1z+a_2)(1+a_1z+a_2z^2)}+\lim_{z\to z_2}\frac{(z-z_2)z}{(z^2+a_1z+a_2)(1+a_1z+a_2z^2)}\\
&=\frac{z_1}{(z_1-z_2)(1+a_1z_1+a_2z_1^2)}-\frac{z_2}{(z_1-z_2)(1+a_1z_2+a_2z_2^2)}\\
&=\frac{z_1(1+a_1z_2+a_2z_2^2)-z_2(1+a_1z_1+a_2z_1^2)}{(z_1-z_2)(1+a_1z_1+a_2z_1^2)(1+a_1z_2+a_2z_2^2)}\\
&=\frac{1-a_2z_1z_2}{(1+a_1z_1+a_2z_1^2)(1+a_1z_2+a_2z_2^2)}\\
&=\frac{1-a_2z_1z_2}{1+a_1(z_1+z_2)+a_2(z_1^2+z_2^2)+a_1^2z_1z_2+a_1a_2z_1z_2(z_1+z_2)+a_2^2z_1^2z_2^2}\\
&=\frac{1-a_2z_1z_2}{1+a_1(z_1+z_2)+a_2[(z_1+z_2)^2-2z_1z_2]+a_1^2z_1z_2+a_1a_2z_1z_2(z_1+z_2)+a_2^2(z_1z_2)^2}
\end{aligned} \tag{13B.4}$$

现在，根据 13B.3 式，

$$z^2+a_1z+a_2=(z-z_1)(z-z_2)=z^2-(z_1+z_2)z+z_1z_2$$

因此

$$a_1=-(z_1+z_2)$$

$$a_2=z_1z_2$$

利用 13B.4 式中 a_1 和 a_2 的值，我们得到

$$s_1^2 = \frac{1 - a_2^2}{1 - a_1^2 + a_2(a_1^2 - 2a_2) + a_1^2 a_2 - a_1^2 a_2^2 + a_2^4}$$

$$= \frac{1 - a_2^2}{1 - a_1^2 - 2a_2^2 + 2a_1^2 a_2 - a_1^2 a_2^2 + a_2^4}$$

$$= \frac{1 - a_2^2}{(1 - a_2^2)^2 - a_1^2(1 - 2a_2 + a_2^2)}$$

$$= \frac{1 - a_2^2}{(1 - a_2^2)^2 - a_1^2(1 - a_2)^2}$$

$$= \frac{1}{(1 - a_2^2) - a_1^2(1 - a_2)/(1 + a_2)}$$

所以

$$s_1^2 = \frac{1}{(1 - a_2^2) - a_1^2(1 - a_2)/(1 + a_2)} \tag{13B.5}$$

第14章　应用和设计研究

本章的目的有四个。第一个目的是描述一些廉价的电路板，可以用于实现在前面章节中描述的DSP算法。我们描述了两个第一代和第二代定点DSP处理器的廉价电路板，利用它们来向学生演示DSP的基本原理。同时还提供了很多第三代定点DSP处理器的纵览。

第二个目的是以研究案例的形式描述了很多DSP的真实世界应用。这里描述的应用包括实时音频信号处理，人类脑电图（大脑的电活动）中伪像（artefact）的自适应滤波，以及胎儿心电图（心脏的电活动）中胎儿心跳的检测，这在分娩过程中估计胎儿的状态时是很有必要的。这部分的介绍用到了很多前面章节中讨论过的DSP概念。

第三个目的是介绍很多具有挑战性的实际问题，在设计学习中将这些问题按照类型进行介绍。最后一个目的是给出一组多项选择题，以帮助读者获得对DSP各个方面更深的理解能力。

14.1　实时信号处理评估板

14.1.1　背景

和工程的其它领域一样，设计和实现DSP算法的实际经验对于正确评估DSP所涉及的问题是很有必要的。只有模拟信号处理背景的工程学学生在掌握DSP所涉及的技术时会遇到真正的困难，特别是如果他们没有必要的数学背景来从理论的观点去理解这些概念。他们经常会被一些问题所迷惑，例如FIR或IIR滤波器中使用的数字操作是如何进行滤波的。他们对模拟滤波器以及把电阻与电容、电感组合在一起的频率特性如何实现滤波的概念相当熟悉。有些人会问，数字滤波器到底是怎么工作的。

我们确信我们需要的是一个简单独立的硬件，学生可以用来设计和实现简单的DSP函数。我们也想验证一些实时DSP涉及到的实际问题，比如混叠的概念、成像、sin x/x、溢出等。这需要开发很多第一和第二代定点DSP处理器的简单目标板。这些电路板仍旧是验证实时DSP算法的有用且不太昂贵的平台。现在，很多第二和第三代定点DSP处理器的廉价评估板可以通过商业途径买到。在下面三节中，我们将描述一些这样的电路板。

14.1.2　TMS320C10目标板

TMS320C10目标板是我们的第一个DSP板。它仍旧适合在单机模式下验证简单实时DSP算法的有用目的。该板的主要特征是

- 能够实时地执行简单DSP算法的单机板；
- 单通道模拟输入/输出，量化为8位；
- 允许很容易地修改DSP算法的代码；
- 能够以两种不同的抽样率操作；
- 允许研究混叠和镜像。

系统框图如图14.1所示，由四个主要单元组成，一个第一代TMS320C10数字信号处理器（为系统的心脏），一个8位的ADC/DAC单元，时序电路，以及存储单元。对于高保真系统，8位分辨率是

不够的，但是对于验证DSP基本原理我们发现这是足够的。存储单元由一个程序选择开关和一对EPROM组成，安装在ZIF（zero insertion force）插座上以方便使用。EPROM划分为八块，每块1 k字，通过程序选择开关进行选择。这允许在EPROM中保存八个不同的程序。对于单机操作，使用EPROM是有必要的。有两个用户可选择的抽样频率可用，一个是7.5 kHz，另一个是15 kHz。

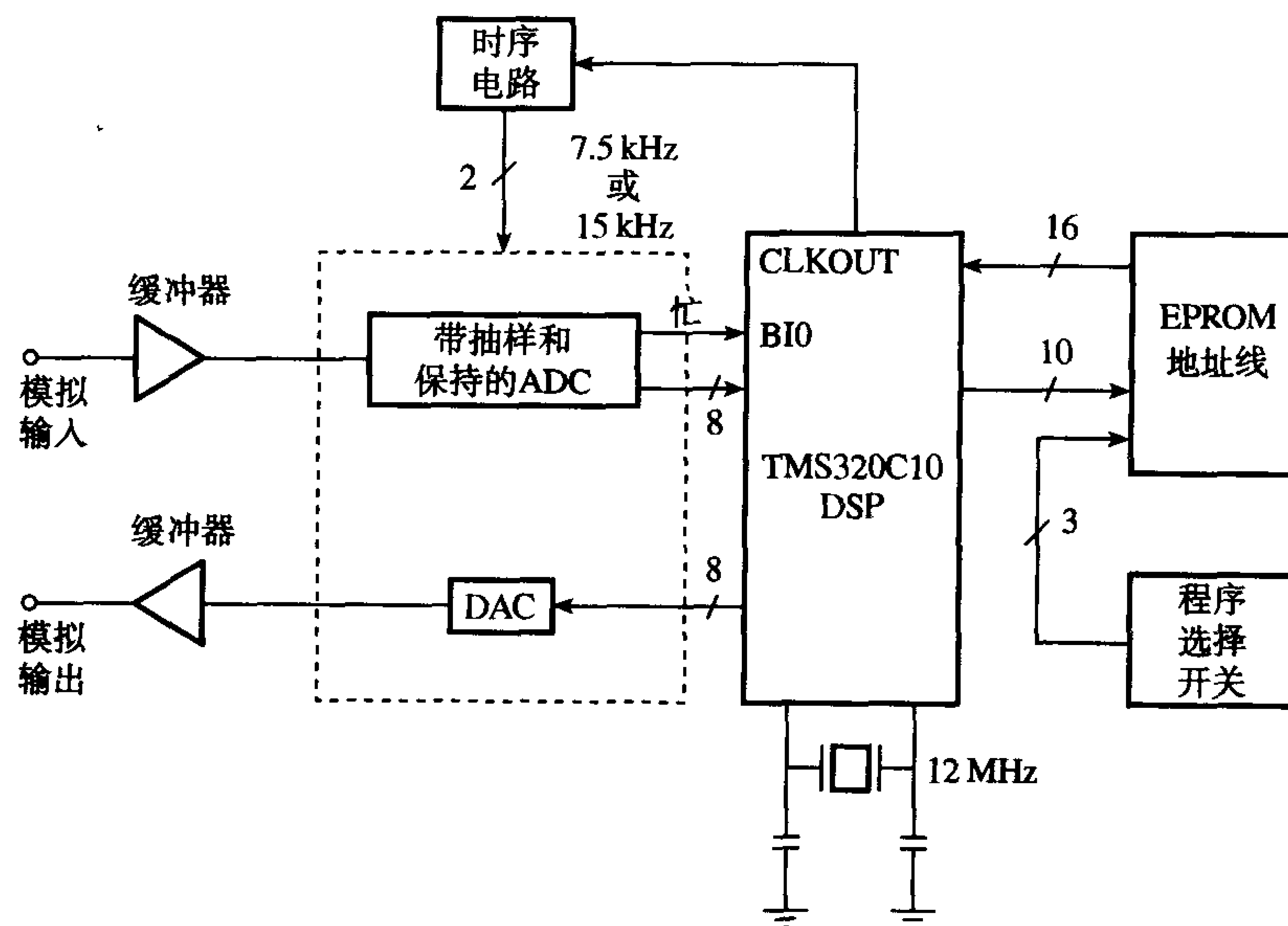


图 14.1　TMS320C10 的简化框图

已经在该目标板上实现了很多的DSP算法。这些算法包括FIR和IIR滤波器、噪声和平方波形产生器。例子在表14.1中列出。

表 14.1　程序存储器选择

块	地址	DSP 程序
0	0000 ~ 03FF	输入－输出循环
1	0400 ~ 07FF	噪声产生器
2	0800 ~ 0BFF	平方波形产生器
3	0C00 ~ 0FFF	41 点带通 FIR 滤波器
4	1000 ~ 13FF	61 点 FIR 陷波滤波器
5	1400 ~ 17FF	串联形式的四阶 IIR 低通滤波器
6	1800 ~ 1BFF	并联形式的四阶 IIR 低通滤波器
7	1C00 ~ 1FFF	串联形式的四阶 IIR 带通滤波器

14.1.3　用于实时 DSP 的 DSP56002 评估模块

TMS320C10板对于在单机模式下验证简单DSP函数是很有用的，但是在重大设计任务中它是受限制的。摩托罗拉DSP56002EVM是一个廉价的评估模块（EVM），它对于快速设计和验证实时DSP系统很有用。在过去的六年中，我们一直在一门DSP课程中使用该EVM，部分原因是DSP56002非常适合于真实音频信号处理这个重点。该EVM的特征包括：

- 一个24位定点DSP56002处理器；
- 32 k字的SRAM和可选择的用于单机操作的32 k字节的flash EEPROM；
- CD质量的音频编解码器（16位立体声A/D和D/A）；

- 抽样率为48、32、16、9.6或8 kHz；
- 汇编器和调试器。

该DSP处理器有两个48位的X和Y寄存器，它们也可以用做四个24位的寄存器（$X0$、$X1$、$Y0$和$Y1$），两个56位的累加器和一个在信号处理中很有价值的硬件乘法器。调试器允许通过简单的屏幕编辑来改变寄存器和源代码。在设计学习部分，将通过一个滤波器设计问题来解释DSP56002板的使用。

14.1.4 TMS320C54和DSP56300评估板

现在已经有了针对新一代的定点和浮点DSP处理器的复杂的软件和硬件开发工具（例如德州仪器的Code Composer Studio），这些工具的详细资料可以在主要厂商的网站上找到。在本节，我们将简要描述两个廉价的评估模块，它们非常适合于学习DSP概念和开发相当高级的DSP系统。

TMS320C54x评估模块（Texas Instruments, 1995）是一个基于PC的插卡，可以用于实时地实现DSP算法。该EVM的主要特征是

- 一个TMS320C541 16位定点DSP处理器，带有5 k字节的片上程序/数据RAM和28 k字节的片上ROM；
- 一个图形的、基于Windows的调试器；
- 对C源码调试器的嵌入式仿真支持；
- 一个模拟I/O接口。

模拟I/O接口支持可编程的抗混叠（anti-aliasing）和抗图像（anti-image）滤波，以及可编程的幅度控制和可编程的抽样率（直到43.2 kHz）。它还提供了一个单通道14位的模拟-数字/数字-模拟转换器。

DSP56302 EVM（Motorola, 1996）是一款优秀的、廉价单机的、链接PC的DSP系统开发平台。用户开发的软件可以从PC下载到片上存储器来执行和调试。DSP56302 EVM的主要特征包括：

- 一个DSP56302的24位定点DSP处理器；
- 板上32 k字程序/高速缓存和数据RAM；
- 两通道CD质量的音频编解码器（16位ADC/DAC）；
- 交叉汇编器和基于Windows的调试器。

14.2 DSP应用

14.2.1 分娩过程中胎儿心跳的检测①

在世界范围内，在分娩期间监测胎儿的标准方法是显示连续的胎儿心率（FHR）和子宫活动，二者一起构成心动图（CTG）（参见图14.2）。通过分析并适当地解释CTG中的变化，产科医生希望防止因为在分娩和接生过程中的缺氧而造成死亡或受损胎儿的出生。

14.2.1.1 胎儿心电图

胎儿心率通常是在分娩时从心电图（ECG）、心脏的电活动（参见图14.3）或超声波中得到的。和成人ECG一样，正常的胎儿ECG也是由五个波峰和波谷表征的，用字母表中连续的字母P、Q、R、S和T标识（Greene, 1987）。因此说ECG是由P波、QRS复杂波和T波组成（Greene, 1987）。

① 本节的内容是基于Iead Rezek的研究项目编写的。

如图 14.3 所示，心跳周期也就是 R 峰和 R 峰之间的时间间隔（以毫秒为单位）的倒数，乘以 60 000 给出瞬时心率。图 14.2 上半部的 FHR 模式是连续瞬时心率的一个图示。

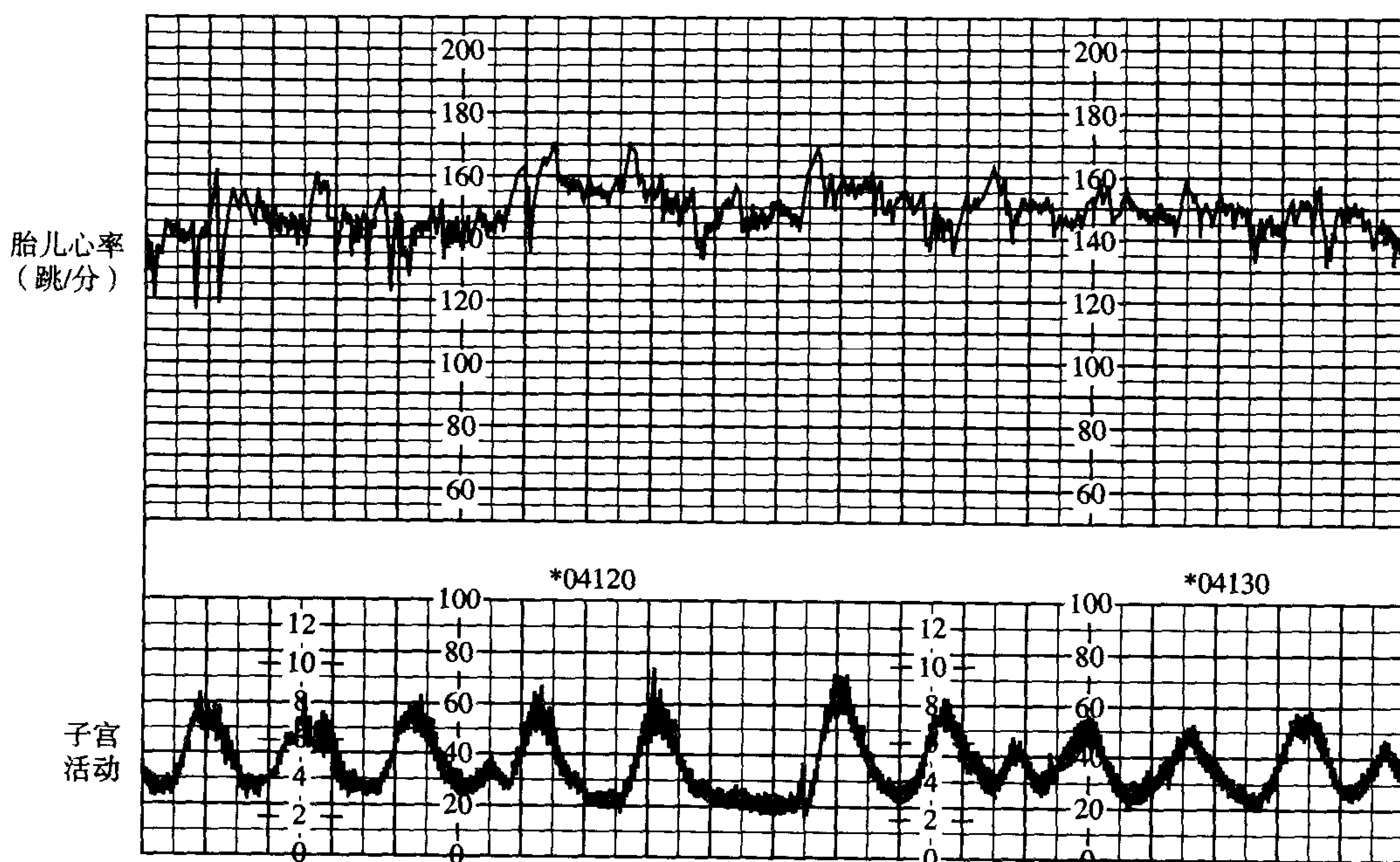
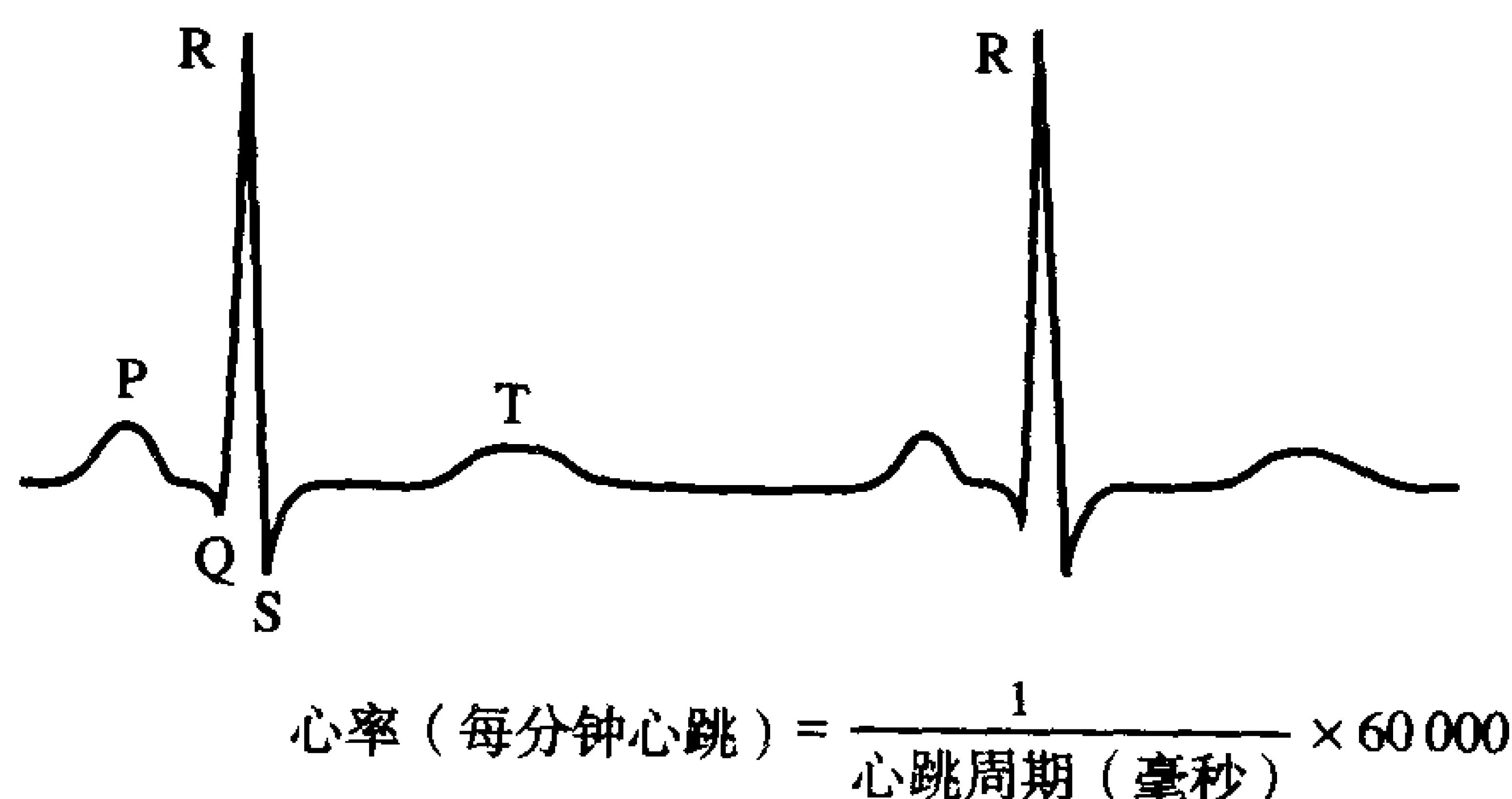


图 14.2　一个 CTG 的例子。CTG 由胎儿心率模式和子宫活动组成



$$\text{心率（每分钟心跳）} = \frac{1}{\text{心跳周期（毫秒）}} \times 60\,000$$

图 14.3　心电图

实际上，为了测量胎儿心率，需要使用一个合适的硬件或软件 DSP 算法来检测连续的 QRS 复杂波，从中计算出 R 到 R 之间的间隔和对应的 FHR。大多数 QRS 检测方法假定胎儿 QRS 复杂波的形状是事先已知的，但是它发生的时间是未知的。这个假定是合理的，尽管不总是有效的，因为 QRS 复杂波的形状可能会随着病人而改变，甚至同一个病人也会改变。因此，通过将 ECG 信号和已知的代表性的 QRS 模板相比较，根据一些相似性的度量，例如高互相关系数，就可以确定 ECG 中 QRS 复杂波的位置。

一个基本的问题是 QRS 复杂波的可靠检测。例如由于基线偏移（baseline wander）、电源干扰（mains interference）、子宫收缩、ADC 饱和及胎儿或母亲的运动而造成的信号恶化，这将导致虚假检测或丢失 QRS 复杂波。这个学习案例的目标是研究和比较两种在实时胎儿监测中可能具有实际价值的 QRS 检测方法。这个工作是一项正在进行的本地医院发起的研究的一小部分，目的在于开发智能系统以帮助分娩时繁忙的临床医生（Ifeachor et al., 1991）。

在学习案例中使用的胎儿ECG数据来自我们的胎儿研究数据库。ECG信号是通过一个放在胎儿头皮上的电极和一个放在母亲大腿上的标准皮肤电极之间的差分测量而得到的，母亲身上的第二个电极用做信号（参见图14.4）。和标准胎儿头皮电极连接方式的径向平面相比，这个领先的系统在胎儿的纵向平面有一个灵敏度矢量，能够减小由于胎儿的转动而引起的ECG矢量的变化（Lindecrantz et al., 1988）。FECG从病人获得输入，隔离的盒子放大，模拟带通滤波（通带0.07～100 Hz），以每秒500个样本及8位分辨率进行数字化。

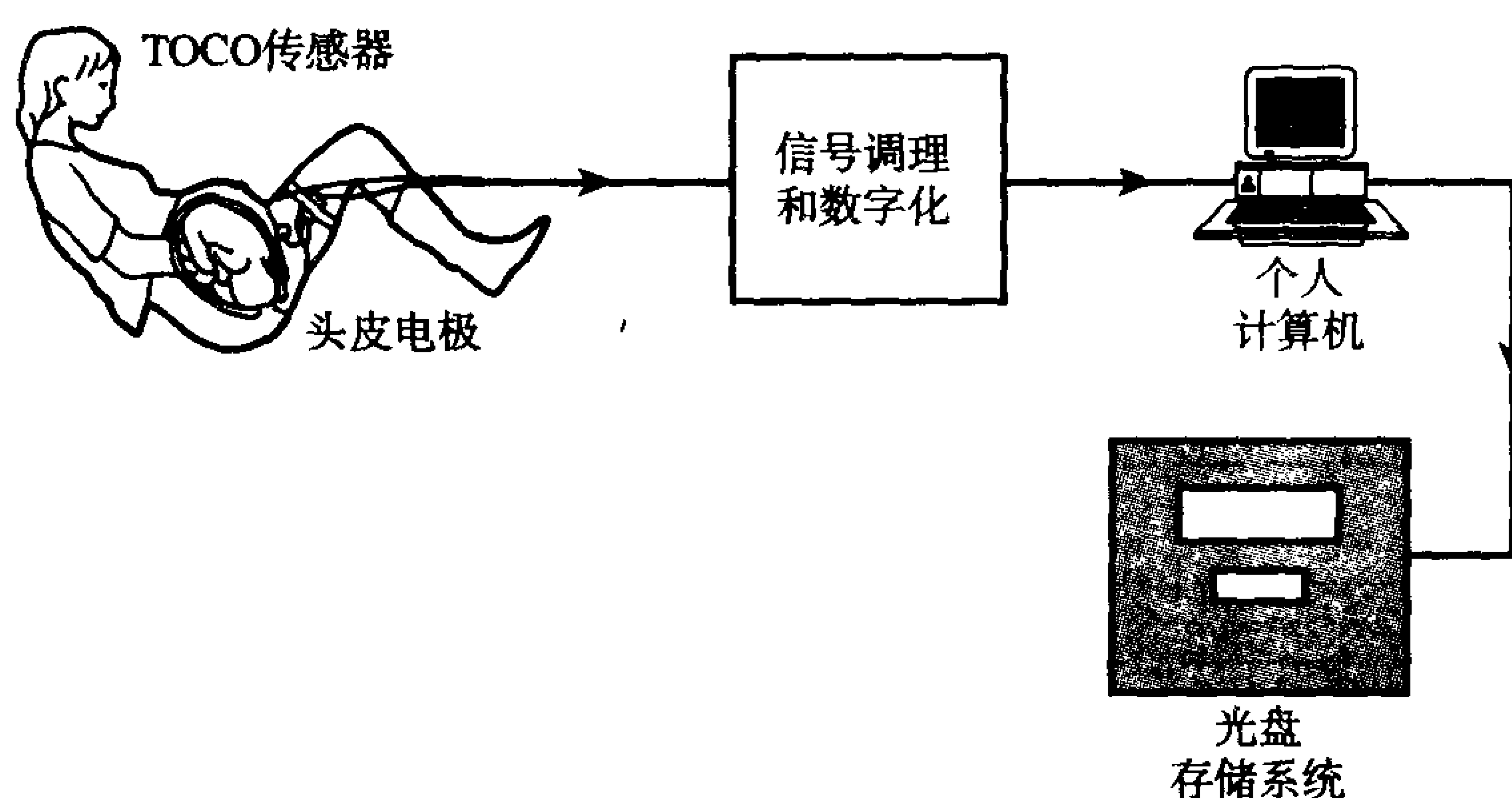


图14.4　胎儿心电图的测量

测量的胎儿ECG的例子如图14.5(a)和图14.5(c)所示。根据视觉检查可以看出，和图14.5(b)和图14.5(c)相比，图14.5(a)中的数据有相对较高的SNR，有大幅度的R波（看成尖刺，spike）。另一方面，图14.5(b)中的数据有相对较高的噪声内容和显著的基线偏移，尽管噪声仍可分辨。图14.5(c)中的数据包含ADC误差，将其看做记录的起始处最大和最小ADC数值之间的大幅度摆动，这大概是由ADC饱和引起的，还有严重的基线偏移和高频噪声（包括电源污染）。图14.5(a)、图14.5(b)和图14.5(c)中的三组数据可以主观地分别划分为1级（好）、2级（良）和3级（差）。

14.2.1.2　胎儿ECG信号预处理

对于2级和3级数据，噪声电平和基线偏移使从原始ECG中检测QRS复杂波更加困难。对于可靠的QRS检测，有必要在试图检测QRS复杂波之前预处理原始ECG以最小化这些信号恶化源的影响。已知QRS复杂波的主要频率成分位于4～45 Hz之间。ECG中的基线偏移一般是低频，通常低于3 Hz，尽管对于3级数据，基线频率可能扩展到15 Hz或更高。

FIR或IIR带通数字滤波器可以用于在QRS检测前预处理原始ECG。我们喜欢使用FIR滤波器，因为高阶（例如八阶）的IIR滤波器在被窄QRS复杂波激励时有时会产生振铃，这可能会使R波的精确位置复杂化。在学习中使用的滤波器指标如下：

滤波器长度	75
抽样频率	500 Hz
阻带	0～1 Hz，47～250 Hz
通带	9～39 Hz
通带波纹	0.5 dB
阻带衰减	30 dB

滤波器系数使用第7章中描述的最佳方法得到。图14.6(a)到图14.6(c)显示了滤波后的ECG数据。和对应的未滤波原始数据相比，图14.6(a)到图14.6(c)的滤波数据中的基线偏移以及高频噪声已经减小

（忽略滤波数据中初始的短暂时间）。在 3 级数据中，作为爆炸形式显现的 ADC 误差无疑会迷惑大多数 QRS 检测算法，参见图 14.6(c)。

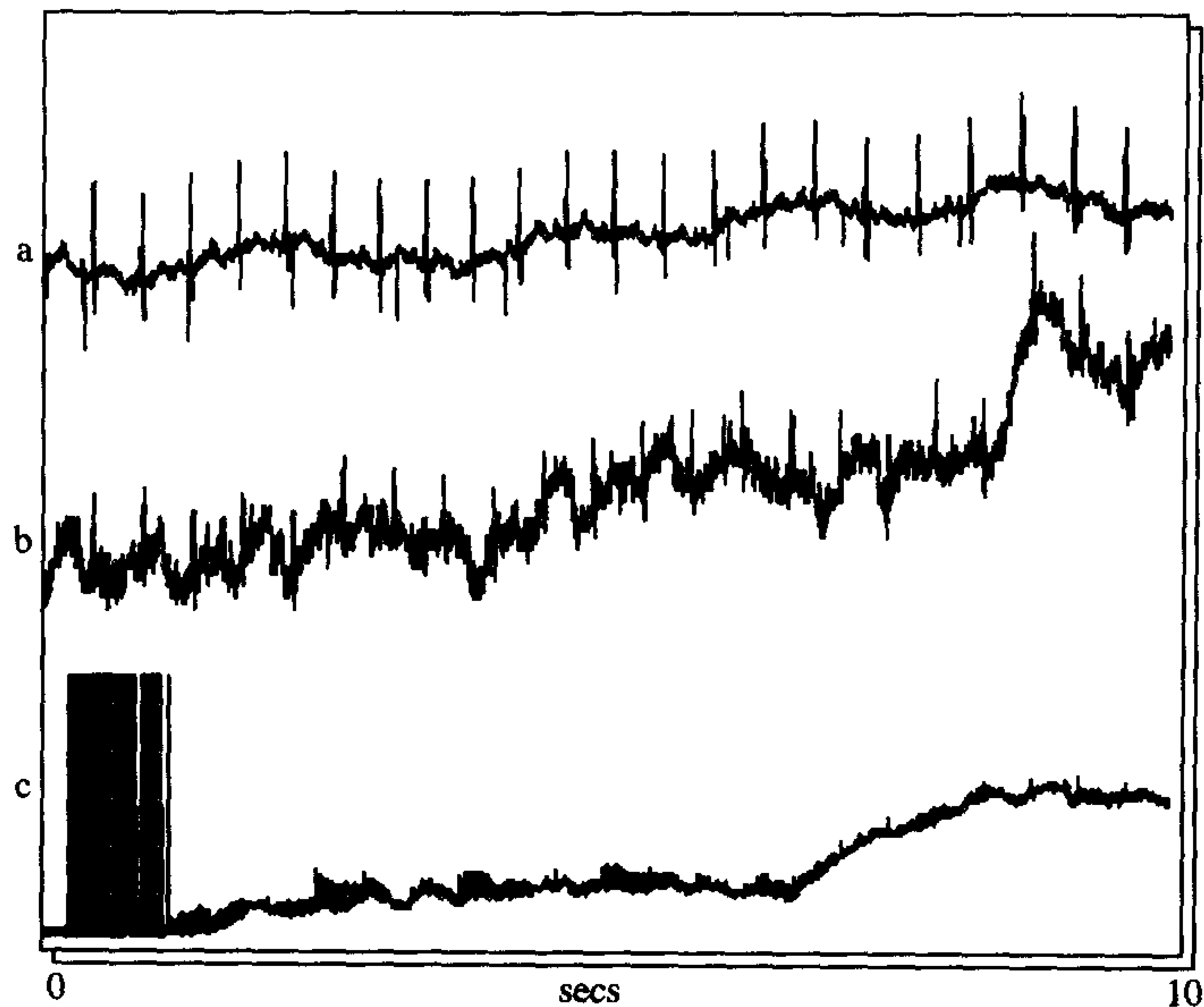


图 14.5　ECG 数据级的例子：(a) 1 级（好）；(b) 2 级（良）；(c) 3 级（差）

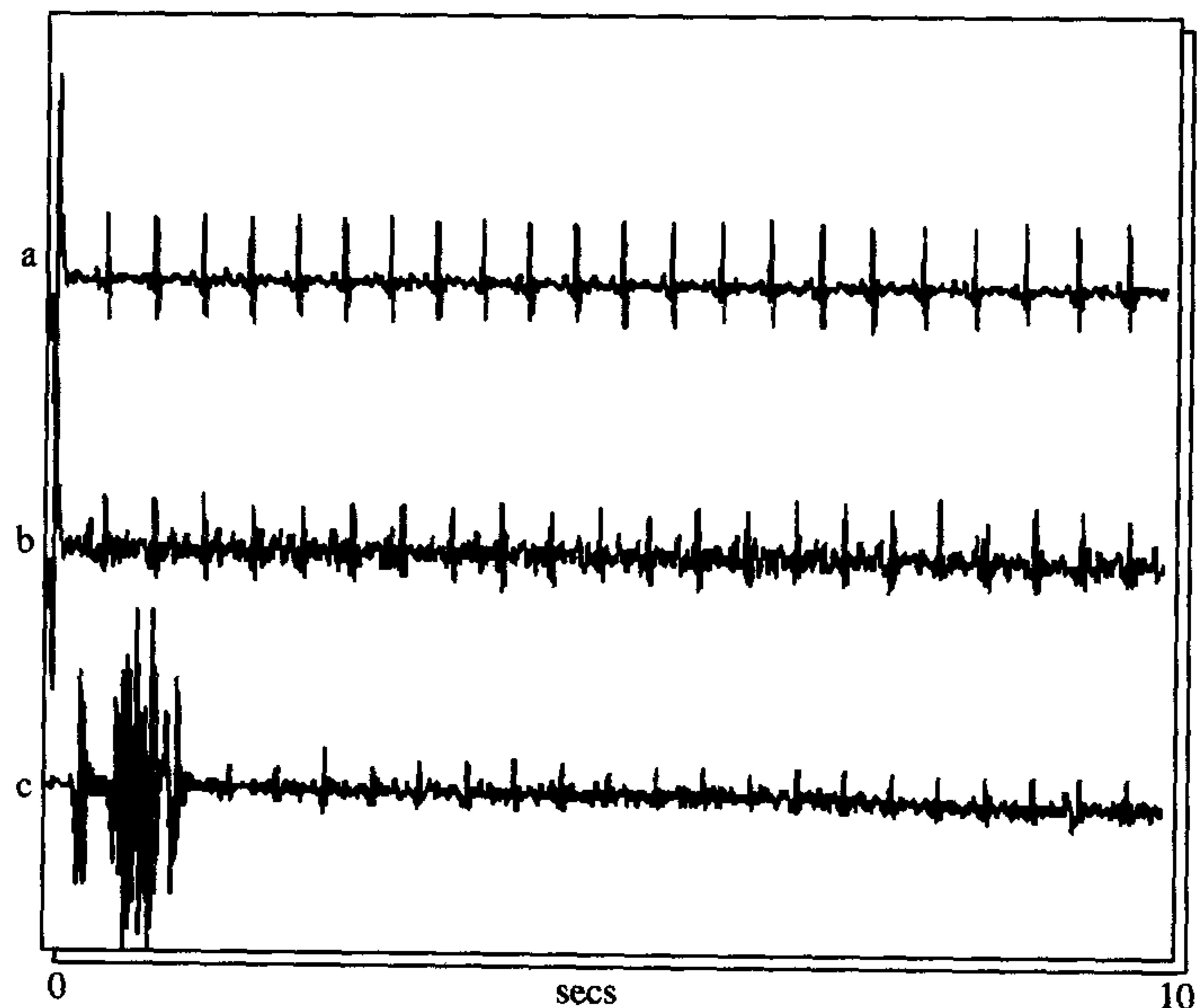


图 14.6　滤波后的 ECG 数据：(a) 1 级（好）；(b) 2 级（良）；(c) 3 级（差）

14.2.1.3　QRS 模板

大多数QRS检测方法依赖于有代表性QRS模板的可用性，引入的ECG信号要与之相比较。模板可以通过检测和平均几个好QRS复杂波来从原始ECG数据产生。这可以通过视觉检查1级ECG记录并且识别好的、不模糊的ECG复杂波而自动或半人工进行。然后同步R波，平均QRS复杂波。一个固定的QRS模板可以用于检测QRS复杂波或者在每个ECG记录的开始处产生一个新的模板。通过平均69个1级数据的QRS复杂波，并且拿出31个平均QRS复杂波样本（R波每边15个样本）的例子如图14.7所示。

在学习中，我们尝试了各种长度的模板。通常，模板长度N在11和31个样本之间，即每秒500个样本的抽样率下大约20 ms到60 ms之间的宽度。在此报告中，我们将给出11个和31个样本两种模板长度得到的结果。

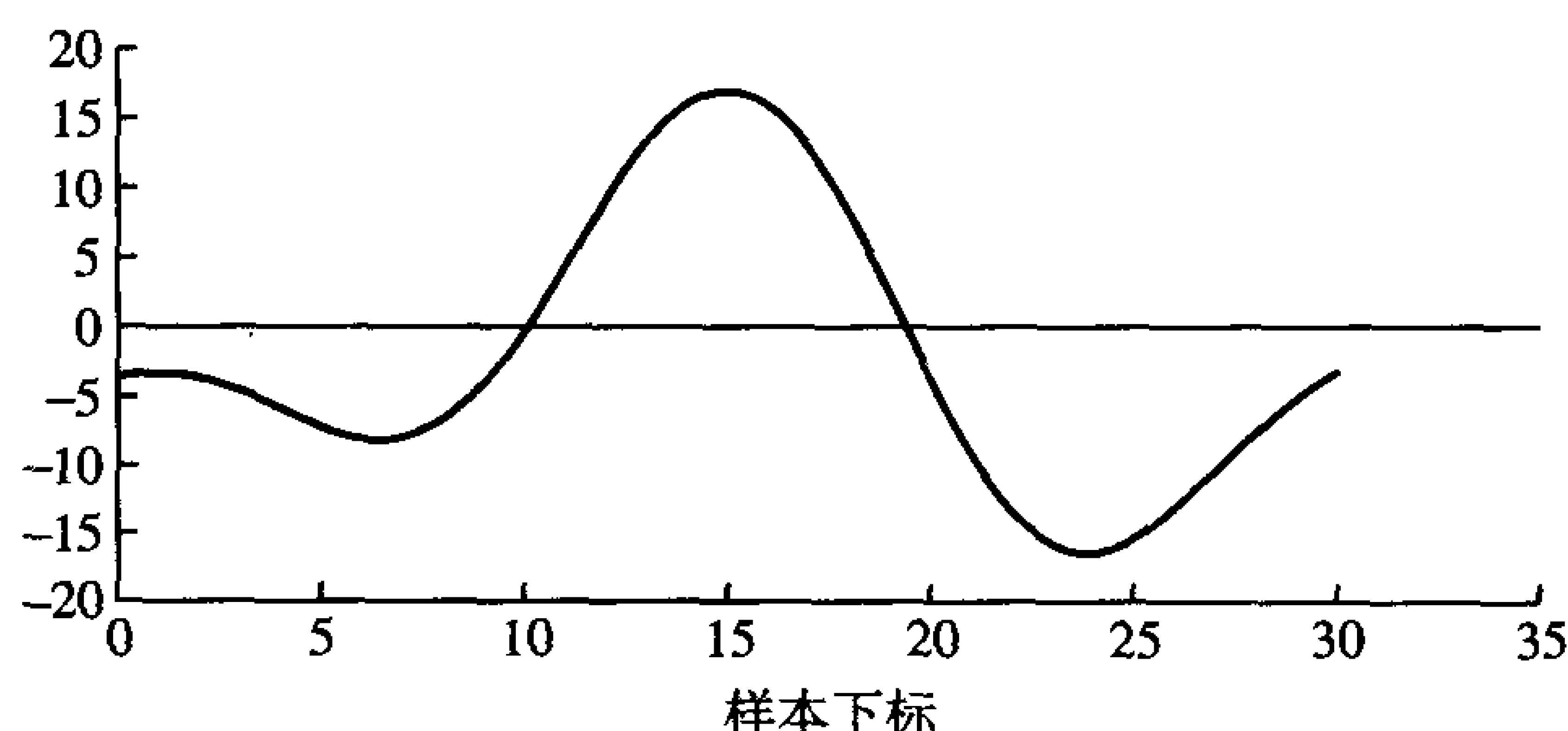


图 14.7 一个QRS复杂波模板的例子。这是通过平均69个R波同步的1级ECG的QRS复杂波得到的。QRS复杂波使用门限电平13检测

14.2.1.4 QRS检测方法

一个通用的QRS检测处理框图在图14.8中给出。原始ECG数据先预处理以减小噪声的影响。预处理后的数据样本每次一个数据点输入一个缓冲器。对输入到缓冲器中的每个新数据点，最早的数据点移出，缓冲器的内容和QRS检测器中的QRS模板进行比较。然后对QRS检测器的输出进行门限比较。如果这个输出超过了门限值，那么就说出现了一个QRS。两个常规的QRS检测方法在实践中进行了比较，选择的依据是它们或者是实际使用的，或者有实际使用的潜力。

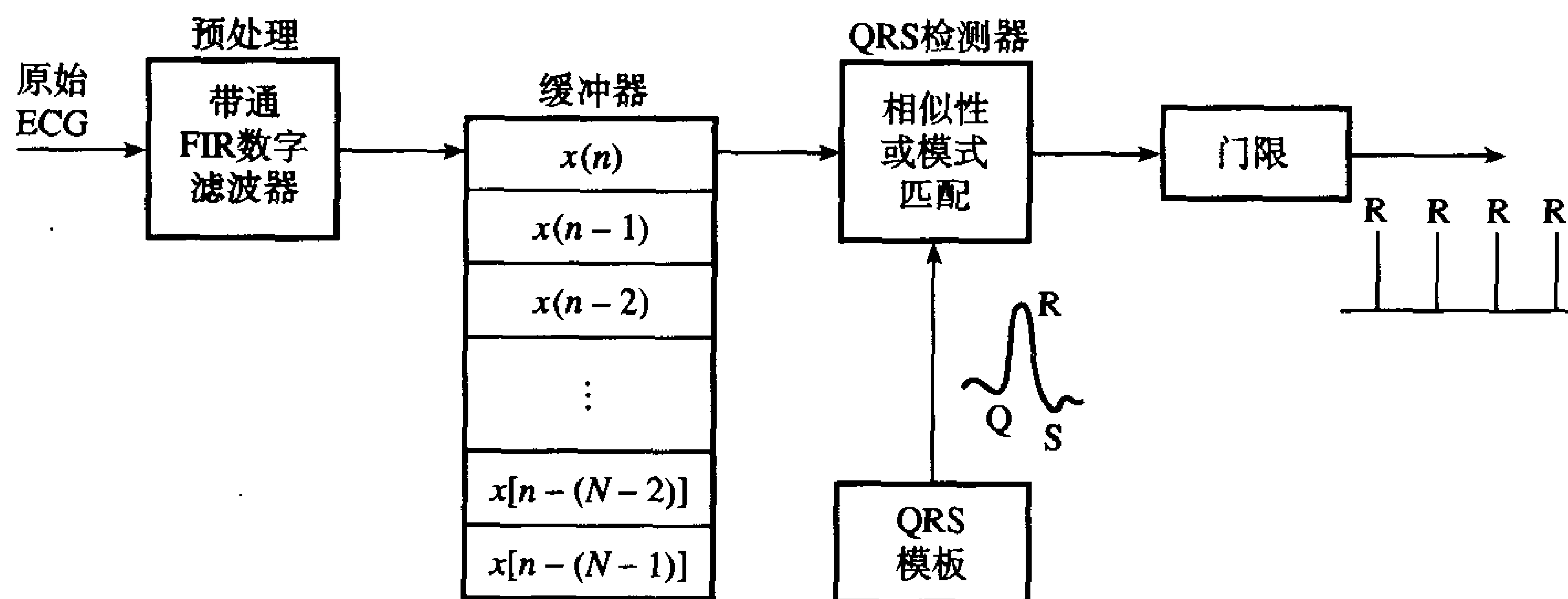


图 14.8 从原始ECG中检测QRS模板的概念

这两个方法是

(1) 平均幅度互差分（average magnitude cross-difference）(AMCD)（Lindecrantz et al., 1988），当前正在Lindecrantz et al.(1988)中描述的一个新型胎儿监测仪中使用；

(2) 匹配滤波，这是一个流行的QRS检测方法，已经经过了很多工作者的研究（Azevedo and Longini, 1980; Favret, 1968）：它和相关方法紧密相关。

平均幅度互差分

在此方法中，预处理后的胎儿ECG数据块和上面描述的一个QRS复杂波模板进行比较。通过波形相减计算ECG和模板对应样本的差分。然后计算差分的绝对值的和$y(i)$：

$$y(i)=\sum_{k=0}^{N-1}\left|x_t(k)-x_t-[x(k+i)-x_i]\right|,\quad i=0,1,\ldots \tag{14.1}$$

这里 $x_t(k)$是 QRS 复杂波模板的样本，$x(k+i)$是 ECG 信号的样本，N 是模板的长度，i 是时移变量。QRS 模板的均值 x_t 和 ECG 信号第 i 个数据块的均值 x_i 如下给出：

$$x_t = \frac{1}{N}\sum_{k=0}^{N-1} x_t(i)$$

$$x_i = \frac{1}{N}\sum_{k=0}^{N-1} x(k+i)$$

当ECG信号和QRS模板形状非常相似时，即在一个QRS复杂波的邻域，AMCD值$y(i)$变为最小（理论上是零）。

数字匹配滤波

匹配滤波一般用于检测隐藏在噪声中的时间重现信号。这个方法主要的根本假设是信号是时间有限的并且有已知的波形。于是问题就是确定它发生的时间。数字匹配滤波器的冲激响应 $h(k)$是要检测信号的时间翻转（time-reversed）的复制。因此在我们的例子中，如果 $x_t(k)$是 QRS 模板，那么匹配滤波器的系数为

$$h(k) = x_t(N-k-1), \quad k = 0, 1, \ldots, N-1 \tag{14.2}$$

数字匹配滤波器可以表示为一个普通横向结构的 FIR 滤波器，滤波器的输出和输入的关系为

$$\begin{aligned} y(i) &= \sum_{k=0}^{N-1} h(k)x(i-k) \\ &= \sum_{k=0}^{N-1} x_t(N-k-1)x(i-k) \end{aligned}$$

这里 $x(i)$是输入 ECG 信号样本，$x_t(k)$是 QRS 模板样本，N 是滤波器长度，$h(k)$是匹配滤波器系数，i 是时移下标。显然当模板和 QRS 复杂波一致时，匹配滤波器的输出将是一个最大值。因此通过搜索门限之上的匹配滤波器的输出，就能够检验 QRS 的发生。

14.2.1.5　QRS 检测的性能度量

为了评估和比较算法，需要一个性能度量。跟随 Azevedo 和 Longini（1980），我们将性能度量定义为

$$\frac{（\text{R 波的总数} - \text{漏掉的数目} - \text{检测失败的数目}）}{\text{胎儿的 R 波总数}} \tag{14.3}$$

对于一个给定的 ECG 记录，只有当记录中所有的 R 波都被正确检测，即没有漏掉（未检测到的 R 波）以及没有虚假检测（虚警）时，性能度量才能达到100%的数值。对于一个给定的 QRS 检测方法，漏检或虚检的次数可以通过视觉比较检测器的输出和预处理后的 ECG 数据确定。另一个可选择的方法是所谓的 28 跳规则，临床医生用来辨别胎儿心率中真正的变化以及如由仪器误差等引起的虚假的变化。根据这个规则，R 波引起的胎儿心率的频率变化超过每分钟±28 跳就意味着一个漏掉的或者虚假的 QRS 复杂波。一个适合于 FHR 模式的基线可以帮助这个规则的应用。

14.2.1.6　结果

图 14.9 和图 14.10 显示了 AMCD 和匹配滤波方法的性能，对比了 1 级和 2 级数据的各种电平。

这两种方法的性能都依赖于使用的门限电平（每个门限表示为最大输入信号的百分比）和 QRS 模板的长度。这两种方法都是在门限电平大约为 50%的时候得到最好性能。当数据质量好的

时候，更宽的模板比窄模板执行得更好，但是它们之间的主要差别好像在于它们对门限电平的灵敏性。

总之，根据它们的性能，在AMCD和匹配滤波方法之间没有多少选择余地。使用合适的门限电平，这两种方法可以达到下列性能：

- 对所有1级ECG 100%的检测；
- 对2级数据>90%的检测；
- 对3级数据>60%的检测。

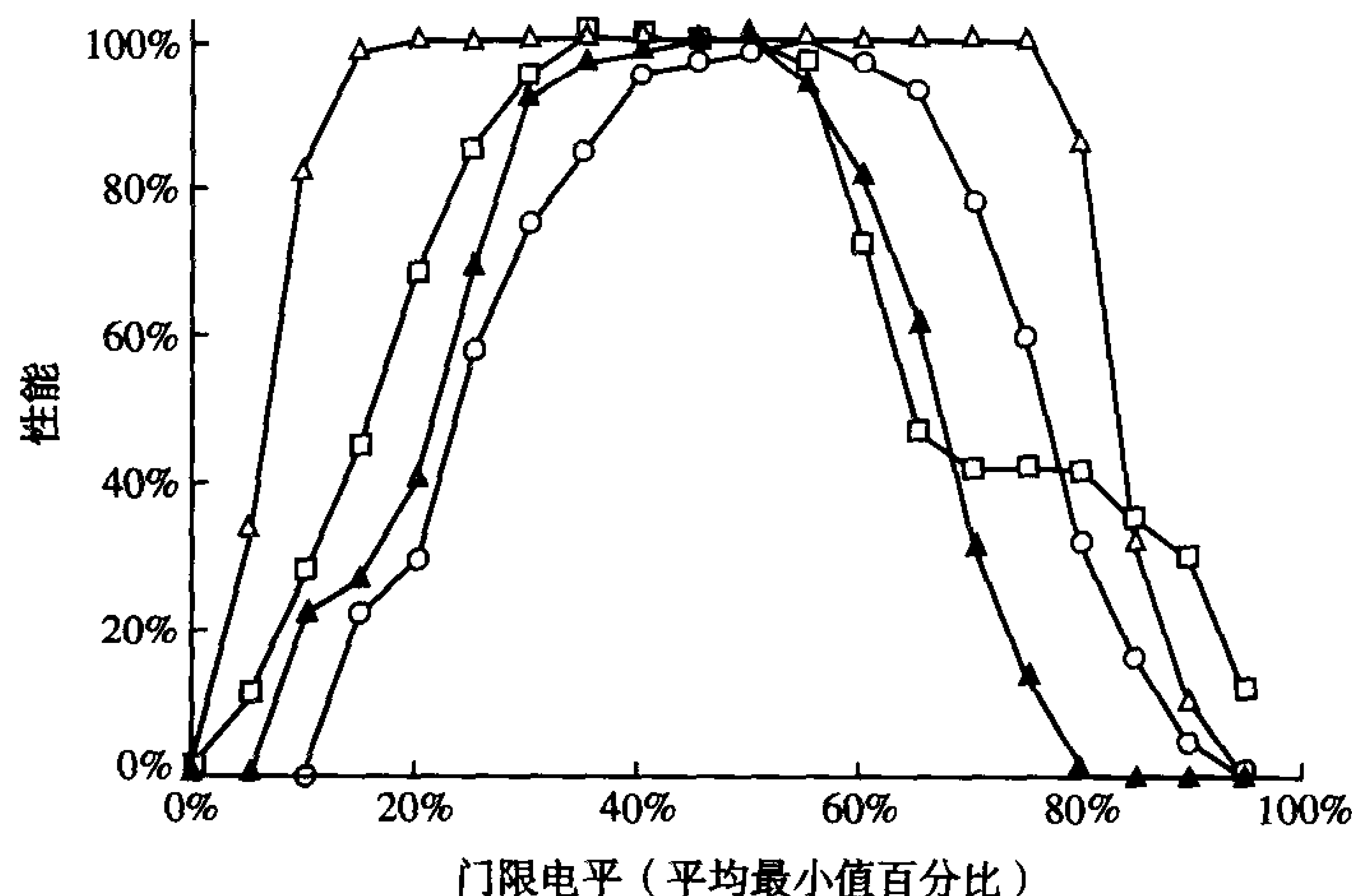


图14.9 对于1级和2级数据，11和31的模板长度，平均幅度互差分方法的性能：-□-，1级，T-11；-△-，1级，T-31；-○-，2级，T-11；-▲-，2级，T-31

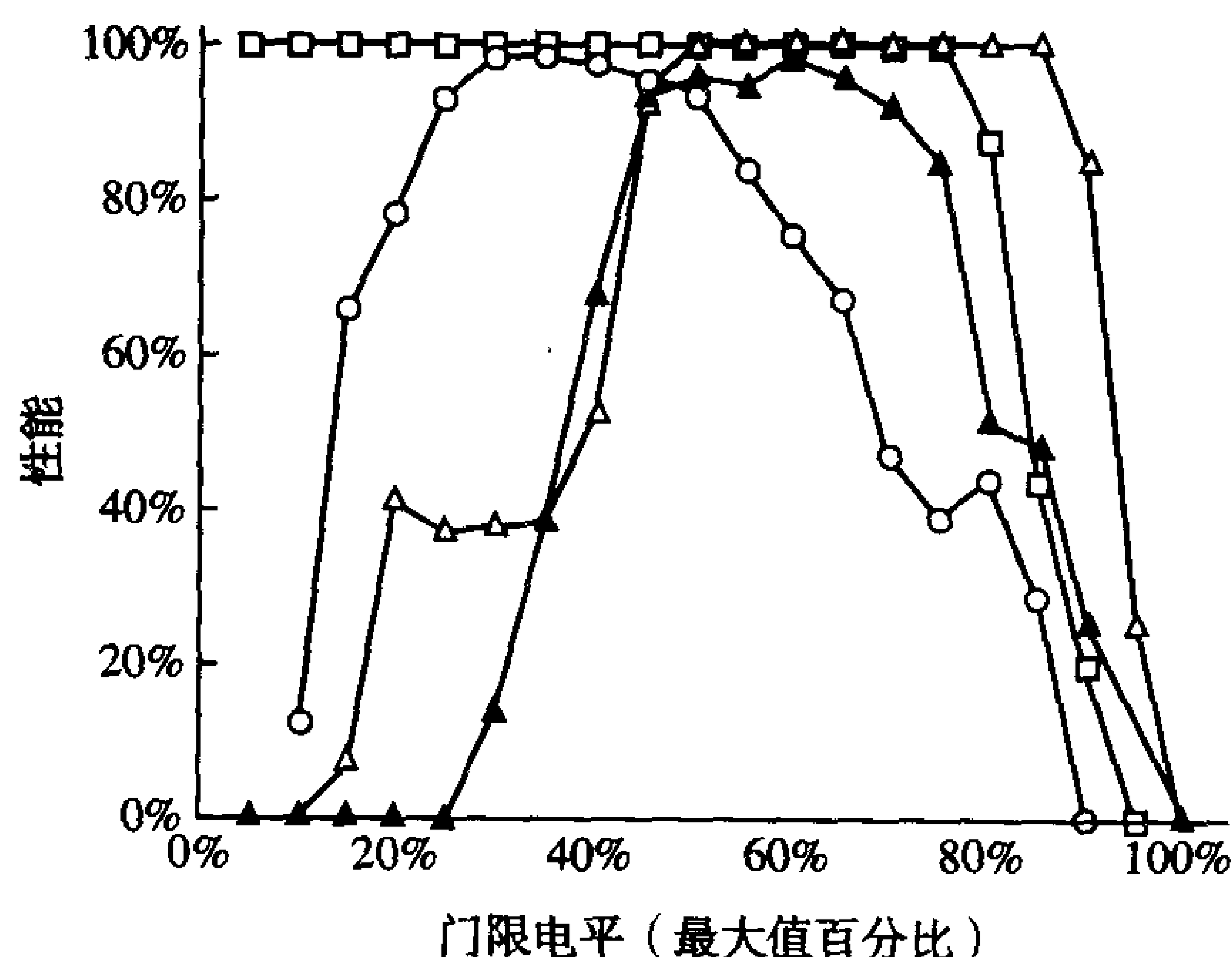


图14.10 对于1级和2级数据、11和31的模板长度，匹配滤波方法的性能：-□-，1级，T-11；-△-，1级，T-31；-○-，2级，T-11；-▲-，2级，T-31

14.2.2 人类EEG中视觉伪像的自适应消除①

14.2.2.1 简介

在本节中描述的工作集中于人类脑电图（EEG）中视觉伪像（OA）的在线消除。EEG广泛用于临床和心理的情况，但是经常被眼系统（眼球、眼睑等）的运动所产生的视觉伪像严重污染。在

① 改编自Ifeachor et al., 1986。

某些情况下，例如前脑有肿瘤的脑部受损的胎儿和病人，很难分辨EEG中与病理相关的慢波和视觉伪像。OA和感兴趣信号之间的相似性也使计算机很难自动分析EEG。一种和刺激相关的响应，称为暂时消极变化（contingent negative variation，CNV），其对患有Huntingdon舞蹈病的患者有诊断用途（Jervis et al., 1984），对于视觉伪像非常敏感。因此有必要从EEG中消除OA，这样才能分析真正的EEG记录。

尽管令人满意的OA消除在离线时是可能的，但是在线OA消除迄今不能令人满意。以前报告的在线方法要求患者的配合，这并不能总是得到保证，同时涉及耗时的人工校准，并且最好只处理一种类型的OA，因为它们假定一个恒定的修正因子。在本节，我们描述了一个新的从EEG信号中消除OA的在线系统，它克服了这些缺点并且提供附加的优点，比如灵活性。该系统基于摩托罗拉68000微处理器，使用数值稳定的UD因子分解算法，允许连续的自适应OA消除。我们介绍了这种在线算法以及该OAR系统硬件和软件的描述。

视觉伪像的消除和控制方法

从EEG消除OA的问题，由于OA信号和一些感兴趣的脑电波之间的相似性，以及它们之间的频谱重叠而变得复杂化。在已经提出的消除或控制EEG信号的OA的各种方法中，眼电图（EOG）相消法可能是最好的。在本章，术语EOG指由于眼部运动造成的电动势，在两个放置在眼附近的表皮电极之间测量。然而，至今报告的各种EOG相消技术并没有完全解决这个问题，人们还在不断地开发新的方法。所有的技术都是根据OA和背景EEG是相加的原理。因此，在离散形式中：

$$y(i)=\sum_{j}^{n}\theta_j x_j(i)+e(i)=\mathbf{x}^{\mathrm{T}}(i)\boldsymbol{\theta}+e(i) \tag{14.4}$$

这里

$$\mathbf{x}^{\mathrm{T}}(i)=[x_1(i)\quad x_2(i)\quad \ldots\quad x_n(i)]$$
$$\boldsymbol{\theta}=[\theta_1\quad \theta_2\quad \ldots\quad \theta_n]^{\mathrm{T}}$$

$y(i)$和$x_j(i)$分别是测量的EEG和EOG的样本，$e(i)$是"真正"的EEG，可以看成一个误差项，i是样本数。θ_j是比例常数，可以称为视觉伪像参数，n是模型中参数的个数。θ_j也称为传输系数。$\mathbf{x}^{\mathrm{T}}(i)$和$\boldsymbol{\theta}$分别是EOG矢量和视觉伪像参数，T表示转置。如果能够估计θ_j，那么就能得到$e(i)$的一个估计为

$$\hat{e}(i)=y(i)-\sum_{j}^{n}\hat{\theta}_j x_j(i),\quad i=1,2,\ldots,m \tag{14.5}$$

这里$\hat{\theta}_j$是θ_j的估计，$\hat{e}(i)$是$e(i)$的估计，m是估计中使用的样本数。问题变成估计θ_j。图14.11解释了这个问题。对于一个给定类型的眼部运动，$\hat{\theta}_j$是相当恒定的，但是在不同类型的OA之间显著不同，尽管有证据表明$\hat{\theta}_j$至少有缓慢的变化，甚至对于一种给定类型的OA。总之，没有方法知道将在一个给定时间发生的OA的类型；而且，因为在很多情况下有不止一种类型的OA同时发生，$\hat{\theta}$不能假定为恒定的，因此最好自适应地估计θ_j。这里使用的术语"自适应"表示OA参数估计，以及由此的OA消除，应该自动地随着OA中的变化进行调节。各种EOG相消技术的不同主要在于θ_j的估计方法，使用的EOG信号的个数以及它们的测量方法。

EOG相消方法可以是在线的，即采集数据的同时，或者是离线的。离线方法比前面报告的在线方法的主要优点是可以采用更复杂的消除技术。然而，在要求实时处理和分析的应用中，采用离线方法所涉及的延迟是不可接受的。EEG信号处理的趋势显然是实时处理，因此在线消除伪像是必要的。

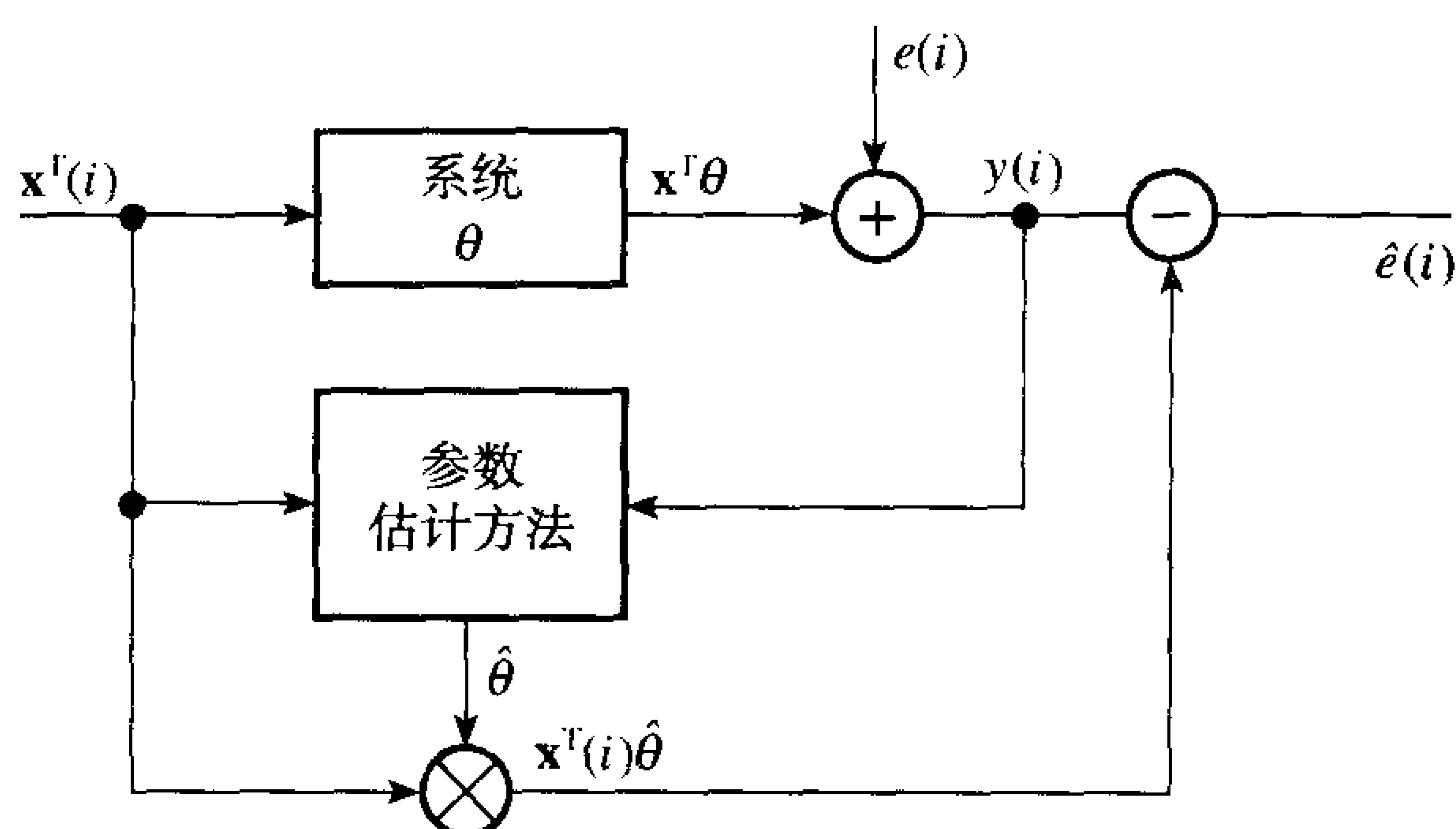


图 14.11 视觉伪像消除的框图表示

在离线方法中，θ的估计是通过最小化J——误差项的平方和得到的，即$J=\sum_{i=1}^{m}e^{2}(i)$。最小化导致

$$\hat{\theta}_m=[\mathbf{X}_m^{T}\mathbf{X}_m]^{-1}\mathbf{X}_m^{T}\mathbf{Y}_m \quad (14.6)$$

这里

$$\mathbf{Y}_m=[y(1)\quad y(2)\quad \ldots\quad y(m)]^{T},\ \mathbf{X}_m=[\mathbf{x}^{T}(1)\quad \mathbf{x}^{T}(2)\quad \ldots\quad \mathbf{x}^{T}(m)]^{T}$$
$$\hat{\boldsymbol{\theta}}_m=[\hat{\theta}_1\quad \hat{\theta}_2\quad \ldots\quad \hat{\theta}_m]^{T},\ \mathbf{E}_m=[e(1)\quad e(2)\quad \ldots\quad e(m)]^{T}$$

这个公式给出了$\boldsymbol{\theta}$的常规最小平方（ordinary least-square，OLS）估计可以使用任何合适的矩阵求逆技术得到，形成了离线OA消除方法的基础。我们已经知道了OA的估计$\hat{\boldsymbol{\theta}}_m$，因此背景EEG的估计$\hat{e}(i)$就能够从14.5式得到。

这里描述的OLS方法可以扩充到多通道的情况，修正超过一个的EEG信号。因此一个带有n个EOG输入和q个测量的EEG输出的系统，可以作为q个单独的单输出子系统处理，整个系统用q种独立的方法识别。

前面报告的在线方法的一个典型例子在图14.12中描述。在此方法中（Girton and Kamiya, 1973），通过在患者重复地于垂直或水平面运动他的眼睛的同时调节电位计，直到在EEG轨迹中有最小量的OA，从而进行初始化校准。然后设备在记录过程中保留这种设置。很多工作者已经使用了这种方法，并且发现它在消除OA中非常难于使用而且效率很低（Gotman et al., 1975）。

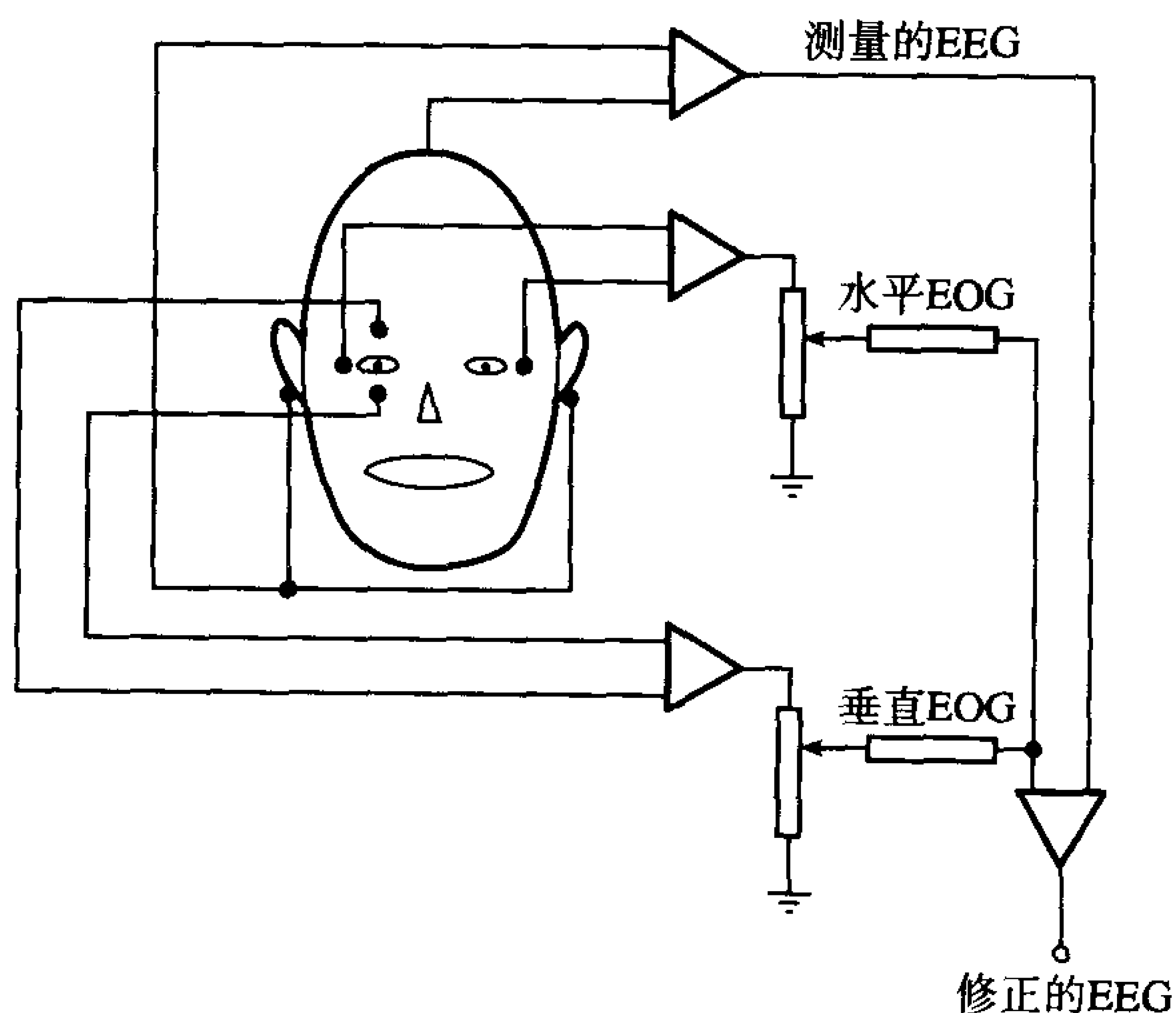


图 14.12 前面报告的消除视觉伪像的在线方法的典型例子（Girton and Kamiya, 1973）

前面报告的采用模拟方法消除视觉伪像的在线方法（Barlow and Rémond, 1981; Girton and Kamiya, 1973），要求患者的配合不能总是得到保证。总体上说，它们在消除OA方面不如离线方法（Jervis et al., 1985; Gotman et al., 1975），要求用户非常熟悉该技术，并且涉及到非常耗时的每个EEG通道的人工校准。另外，这些方法只能适用于一种类型的OA。

在本节，将描述一个能够克服上述缺点的消除视觉伪像的在线系统。它是一个基于微处理器的系统，使用了一种基于有效的递归最小平方技术的数值稳定在线算法。

14.2.2.2 在 OAR 系统中使用的在线消除技术

14.6式中 $\hat{\boldsymbol{\theta}}_m$ 的计算要求非常耗时的逆矩阵计算。显然，OLS 方法不适合于实时或在线估计，因为在线算法涉及到固定的严格数目的算术操作，倾向于没有矩阵求逆。使用在线算法，在每个样本点更新 $\hat{\theta}$，因此眼部运动中的变化能够在 $\hat{\theta}$ 中反映出来，本例中的 OA 消除可以看成是 EEG 中 OA 的自适应滤波（Widrow et al., 1975）。

两种适合于在线估计 $\boldsymbol{\theta}$ 的参数估计方法是最小均方（LMS）（Widrow et al., 1975）和递归最小平方（RLS）算法（Peterka, 1975; Clarke, 1981）。根据计算和存储，LMS 算法比 RLS 算法更有效。另外，它没有 RLS 算法固有的数值不稳定问题（参见后面的内容）。然而，RLS 算法比 LMS 算法具有更好的收敛性，因为这个原因，它受到了普遍欢迎。

LSM 算法的常用形式为

$$\hat{\boldsymbol{\theta}}(m+1) = \hat{\boldsymbol{\theta}}(m) + 2\mu\mathbf{x}(m+1)[y(m+1) - \mathbf{x}^{\mathrm{T}}(m+1)\hat{\boldsymbol{\theta}}(m)] \tag{14.7}$$

这里 $\hat{\boldsymbol{\theta}}(m)$ 和 $\hat{\boldsymbol{\theta}}(m+1)$ 分别是 $\boldsymbol{\theta}$ 在第 m 和第 $(m+1)$ 个样本点的估计，μ 是一个控制收敛速率和算法稳定性的常数。为了收敛，μ 应该在下面的范围之内：

$$0 < \mu < 1/\lambda_{\max}$$

这里 $\lambda_{\max}$ 是14.6式的矩阵（$\mathbf{X}_m^{\mathrm{T}}\mathbf{X}_m$）的最大特征值。然而，算法的收敛时间直接正比于（$\mathbf{X}_m^{\mathrm{T}}\mathbf{X}_m$）最大和最小特征值的比值。当输入变量——在本例中即 EOG 中存在很强的共线性（collinearity）时，这个比值可能很大。然而，LMS 算法广泛用于生物医学应用中，以减小噪声或主要因为相似性而产生的伪像。

一个合适的递归最小平方（RLS）算法通过对数据进行指数加权以逐渐消除老数据对估计的影响而得到。因此

$$J = \sum_{i=1}^{m} \gamma^{m-i} e^2(i), \quad 0 < \gamma < 1 \tag{14.8}$$

J 关于 $\boldsymbol{\theta}$ 的最小化引出下面的递归最小平方算法：

$$\hat{\theta}(m+1) = \hat{\boldsymbol{\theta}}(m) + \mathbf{G}[y(m+1) - \mathbf{x}^{\mathrm{T}}(m+1)\hat{\boldsymbol{\theta}}(m)] \tag{14.9a}$$

$$\mathbf{P}(m+1) = \frac{1}{\gamma}\left[\mathbf{P}(m) - \frac{1}{\alpha}\mathbf{P}(m)\mathbf{x}(m+1)\mathbf{x}^{\mathrm{T}}(m+1)\mathbf{P}(m)\right] \tag{14.9b}$$

这里

$$\alpha = \gamma + \mathbf{x}^{\mathrm{T}}(m+1)\mathbf{P}(m)\mathbf{x}(m+1)$$

$$\mathbf{x}^{\mathrm{T}} = [x_1(m+1) \quad x_2(m+1) \quad \ldots \quad x_n(m+1)]$$

$$\mathbf{G} = \mathbf{P}(m+1)\mathbf{x}(m+1) = \mathbf{P}(m)\mathbf{x}(m+1)/\alpha$$

自变量m用来强调量值是在每个样本点得到的这个事实。γ称为遗忘因子（forgetting factor），防止矩阵$\mathbf{P}(m+1)$随着m的增加而趋向于零（$\hat{\theta}(m+1)$趋向于常数），因此允许跟踪一个缓慢变化的参数。通常，γ在0.98和1之间。更小的值给更新的数据分配了太大的加权，导致剧烈波动的估计。

然而，当RLS算法直接实现的时候，可能会遇到两个主要的问题。如果信号不是“持续激励”（persistently exciting）的，例如当没有眼部运动时，会引起第一个问题，称为“爆炸”（blow-up），会导致14.9b式中$\mathbf{P}$的元素指数增加。因此

$$\lim_{m\to\infty}[P_{ij}(m+1)] = \lim_{m\to\infty}\left[\frac{P_{ij}(m)}{\gamma^m}\right] \to \infty \tag{14.10}$$

然而，由于微小的眼部运动（这总是出现的），以及在EOG通道中经常接收到的其他活动，这个问题在OA消除中不是那么严重。

RLS的第二个问题是它对计算机舍入误差的灵敏性，这将导致一个负定（negative definite）的$\mathbf{P}$矩阵，最终会导致不稳定。对于成功的估计，矩阵$\mathbf{P}$有必要是半正定（positive semi-definite）的，这等效于在离线的情况下要求矩阵（$\mathbf{X}_m^{\mathrm{T}}\mathbf{X}_m$）是可逆的。但是，由于14.9b式中的差分项，$\mathbf{P}$的正定性（positive definiteness）不能保证（Peterka, 1975; Clarke, 1981; Bierman, 1976）。这个问题在多参数模型中更糟糕，特别是如果变量（本例中的EOG）是线性依赖的（Peterka, 1975）以及当算法在一个有限字长的小系统上实现的时候（Clarke, 1981）。

数值不稳定性的问题可以通过适当地因子分解矩阵$\mathbf{P}$以避免14.9b式中的差分项来解决。这样的因子分解算法受到更好的数值条件限制，具有和使用双精度的RLS算法相兼容的准确性（Bierman, 1976）。两种这样的算法是平方根和UD因子分解算法。然而，根据存储和计算，UD算法更有效，因此更受欢迎。实际上，UD算法是常规平方根算法的平方根自由排列（square-root-free arrangement），因此和后者共享同样的性质。

在UD方法中，$\mathbf{P}(m+1)$因子分解为

$$\mathbf{P}(m+1) = \mathbf{U}(m+1)\mathbf{D}(m+1)\mathbf{U}^{\mathrm{T}}(m+1) \tag{14.11}$$

这里$\mathbf{U}(m+1)$是单位上三角矩阵，$\mathbf{U}^{\mathrm{T}}(m+1)$是它的转置，$\mathbf{D}(m+1)$是一个对角线矩阵。因此，替代更新$\mathbf{P}$，它的因子$\mathbf{U}$和$\mathbf{D}$被更新。

使用14.11式，14.9b式可以写为

$$\mathbf{P}(m+1) = \frac{1}{\gamma}\mathbf{U}(m)\left[\mathbf{D}(m) - \frac{1}{\alpha}\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}}\right]\mathbf{U}^{\mathrm{T}}(m) \tag{14.12}$$

这里

$$\boldsymbol{v} = \mathbf{D}(m)\mathbf{U}^{\mathrm{T}}(m)\mathbf{x}(m+1)$$

如果方括号中的项进一步因子分解为一个上三角和对角线矩阵，即

$$\bar{\mathbf{U}}(m)\bar{\mathbf{D}}(m)\bar{\mathbf{U}}^{\mathrm{T}}(m) = \mathbf{D}(m) - \frac{1}{\alpha}\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}} \tag{14.13}$$

这里横线用来区别$\mathbf{D}(m) - (1/\alpha)\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}}$和$\mathbf{P}$中的因子$\mathbf{U}$和$\mathbf{D}$，那么

$$\mathbf{P}(m+1) = \frac{1}{\gamma}\mathbf{U}(m)\bar{\mathbf{U}}(m)\bar{\mathbf{D}}(m)\bar{\mathbf{U}}^{\mathrm{T}}(m)\mathbf{U}^{\mathrm{T}}(m) \tag{14.14}$$

比较14.11式和14.14式，注意到上三角矩阵的乘积是上三角本身以及14.14式的对称性，那么

$$\mathbf{U}(m+1) = \mathbf{U}(m)\bar{\mathbf{U}}(m) \tag{14.15a}$$

$$\mathbf{D}(m+1)=\frac{1}{\gamma}\bar{\mathbf{D}}(m) \tag{14.15b}$$

因此，$\mathbf{U}(m+1)$和$\mathbf{D}(m+1)$的更新问题依赖于为$\mathbf{U}(m)$和$\mathbf{D}(m)$找到合适的递归公式。Bierman(1976)已经为卡尔曼滤波器给出了一个递归地更新$\mathbf{U}(m+1)$和$\mathbf{D}(m+1)$的算法，其中使用了误差项$e(i)$而不是γ的方差。针对OA问题，这个算法进行了微小的修正以合并γ的替代，如上面给出的介绍。修正算法在附录 14A 中给出。

在附录 14A 步骤 10 得到的增益矢量$\mathbf{G}$用来更新参数估计，如 14.9a 式中指出的。因此，尽管$\mathbf{P}(m+1)$能够像在 14.11 式中那样从更新的 UD 元素获得，但是明确地计算$\mathbf{P}(m+1)$是没有必要的。

UD 算法的一些性质

为了得到对UD算法的一些认识，明确地写出该算法是很有用的。因此，对于一个两参数的模型，附录中的算法变成

- 步骤 1：$v_1=x_1(m+1)$; $v_2=x_2(m+1)+U_{12}(m)x_1(m+1)$
- 步骤 2：$b_1=d_1(m)x_1(m+1)$; $b_2=d_2(m)v_2$
- 步骤 3：$\alpha_1=\gamma+b_1v_1=\gamma+d_1(m)x_1^2(m+1)$
- 步骤 4：$d_1(m+1)=d_1(m)/\alpha_1$
- 步骤 5：$\alpha_2=\alpha_1+b_2v_2=\alpha_1+d_2(m)v_2^2$
- 步骤 6：$U_{12}(m+1)=U_{12}(m+1)-b_1v_2/\alpha_1$
- 步骤 7：$b_1=b_1+b_2U_{12}(m)=d_1(m)x_1(m+1)+d_2(m)v_2U_{12}(m)$
- 步骤 8：$d_2(m+1)=d_2\alpha_1/\gamma\alpha_2$

从步骤 3 可以看出，假设对角线元素（即$d_1(0)$和$d_2(0)$）的开始值是正的，α_1和由此的$d_1(m+1)$将总是正的。对于α_2（和 14.9 式中的α相同）和$d_2(m+1)$同样是正确的。

(1) **P 的正定性**　矩阵$\mathbf{P}$是正定的，当且仅当$\mathbf{x}^{\mathrm{T}}\mathbf{P}\mathbf{x}>0$，当$x_1=x_2=\ldots=x_n=0$时除外（Bajpai et al., 1973）。

根据 14.9 式和 14.11 式，$\alpha=\gamma+\mathbf{x}^{\mathrm{T}}\mathbf{P}\mathbf{x}=\gamma+\mathbf{x}^{\mathrm{T}}\mathbf{U}\mathbf{D}\mathbf{U}^{\mathrm{T}}\mathbf{x}$，对于一个两参数模型，其变为（参见上面的步骤 3 和步骤 5）

$$\alpha_2=\alpha_1+d_2v_2^2=\gamma+d_1(m)x_1^2(m+1)+d_2(m)v_2^2$$

所以，在本例中，

$$\mathbf{x}^{\mathrm{T}}\mathbf{P}\mathbf{x}=d_1(m)x_1^2(m+1)+d_2(m)v_2^2$$

因此，可以看出$\mathbf{x}^{\mathrm{T}}\mathbf{P}\mathbf{x}$的符号依赖于对角线元素（即$d_1$和$d_2$）的符号，它总是正的，如前面所声明的，所以矩阵$\mathbf{P}$满足正定性质。因此，UD 因子分解算法保证了$\mathbf{P}$的正定性质。

(2) **爆炸问题**　如果没有数据，即$x_1=x_2=\ldots=x_n=0$，那么$\alpha_1=\alpha_2=\gamma$（步骤 3 和步骤 5），所以d_1和d_2将连续地被γ缩放，它小于1，导致对角线元素的指数增加。因此爆炸问题似乎不能像有时候建议的那样通过矩阵因子分解消除。其他 RLS 方案可以用于减小爆炸的影响（Goodwin and Sin, 1984），但是在 OA 工作中，微小的眼部运动和其他固有的系统噪声确保x的值决不会无限地变为零。

使用UD和平方根算法的在线OA消除已经可以由计算机实现，并且给出了和离线方式相类似的结果。

14.2.2.3 在线视觉伪像消除系统的硬件

在本节和下节，描述了使用UD算法的在线视觉伪像消除（OAR）系统。首先，设置了系统的目标规范。接下来，在系统和框图级描述了一个合适的系统。

OAR 系统是用下面需要注意的需求进行设计的：

(1) 和标准 EEG 机器兼容；

(2) 能够提供连续实时的多通道 EEG 信号中的 OA 消除（OA 消除现在应该是根据主观准则并且应该是自适应的）；

(3) 能够输出修正的 EEG 和 / 或未修正的 EEG 和 EOG 到 EEG 机器，以允许修正和未修正 EEG 的瞬时比较；

(4) 能够避免饱和，这将会减小修正器的有效性，系统应该具有某些自动范围调整功能；

(5) 仪器应该适合不熟练的人员使用。

这些需求可以通过一个合适的基于微处理器的、实现了UD算法的系统而得到满足。使用基于微处理器的仪器还提供了下列优点：

(1) 软件控制的设计产生一个非常灵活的系统。可以在一个系统上实现几个OAR算法和模型，在任何应用中使用的模型由用户规定。

(2) 只需要修改软件就可以研究新的模型或思想，而不需要建造新的仪器。因此软件控制的 OAR 系统是一个优秀的研究辅助工具。

(3) 可编程仪器允许提供自行操作例程（housekeeping routine），用于自检、自动校准、减少负载问题等。消除系统的数据处理可能包括EOG的数字滤波以减小次要伪像的影响（Hamer et al., 1985）。

目前，OAR 系统只能处理六通道 EEG 和 EOG 信号，但是可以扩展到 20 通道（参见后面的内容）。系统的框图在图 14.13 中给出。来自 EEG 机器辅助输出端的每个 EEG/EOG 信号首先放大，然后通过馈给抽样保持电路的低通滤波器限带到 30 Hz。EEG/EOG 信号然后同时在抽样信号 FS 的正跳变（positive transition）抽样。采取同时抽样是为了避免在对应的时间点之间引入延迟。抽样信号 FS 的负跳变（negative transition）然后中断处理器，预示着一个周期的开始，在此周期中多路选择器（MUX）顺序地选择 20 个样本，在微处理器（μP）的控制下由模数转换器（ADC）进行数字化。

可编程增益放大器（PGA）和窗口检测器用于扩展ADC的动态范围以避免过载。ADC过载是OA工作中的一个问题，可能导致虚假的参数估计。为了避免这个问题，普通的做法是只用ADC动态范围的一部分，这样过载就很少，然后丢弃发生过载的区域的所有数据，但是这会导致数据浪费。使用PGA和窗口检测器允许动态地改变增益。因此，当PGA的输出超出预定义的窗口限制时，PGA的增益设置为一个较低的数值，它自动地将样本值减半，将其带到 ADC 的动态范围之内，因此避免了饱和。在数字化的样本存入存储器之前，必须考虑到这种情况。

数字化的样本接着由OAR算法处理以得到修正的EEG。修正的EEG样本，如果需要，可以和原始的 EEG/EOG 一起，通过数字 / 模拟转换器和相关网络输出给 EEG 机器的辅助输入端。

所需的多路复用器和多路分配器允许输入和输出通道的资源共享。一种可选择的方法是为每个通道提供独立的ADC和DAC。这种方法将通道之间交调（cross-talk）引起的系统噪声减小到最小，但是被认为是非常昂贵的。

每个通道有一个独立的带有独立抽样信号线的输出抽样保持。抽样保持用于保存模拟样本直到得到下一个模拟样本。这伸展了样本脉冲，增加了信号功率，但是引入了孔径失真（aperture distortion），在本例中我们认为它很小。

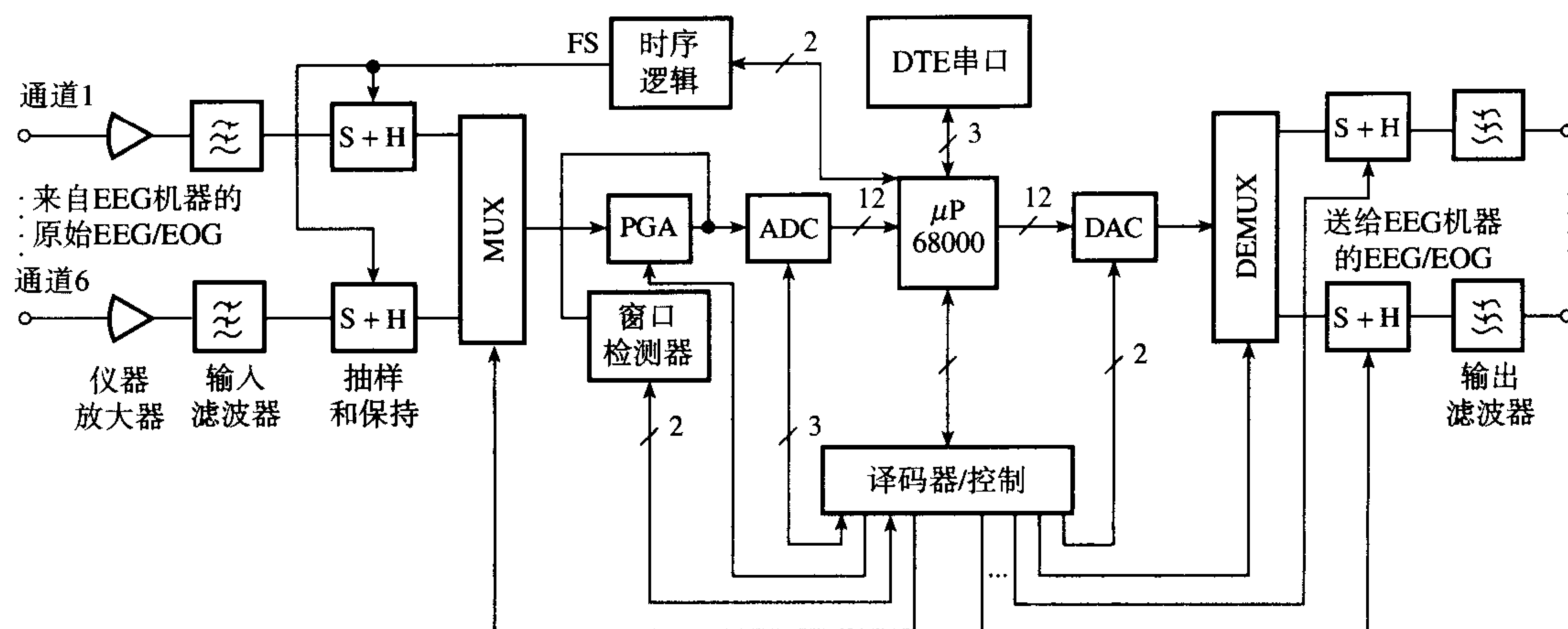


图 14.13　基于微处理器的视觉伪像消除系统的框图

14.2.2.4　在线视觉伪像消除系统的软件

OAR 系统软件由数据采集和发布例程、在线 OA 消除例程、软件浮点算术例程和监控主程序组成。整个软件占据3 k字节的存储器。OAR系统软件的心脏是用68000微处理器汇编语言编写的，即在 14.2.2.2 节中描述的 UD 算法。这里我们将给出软件的概览。

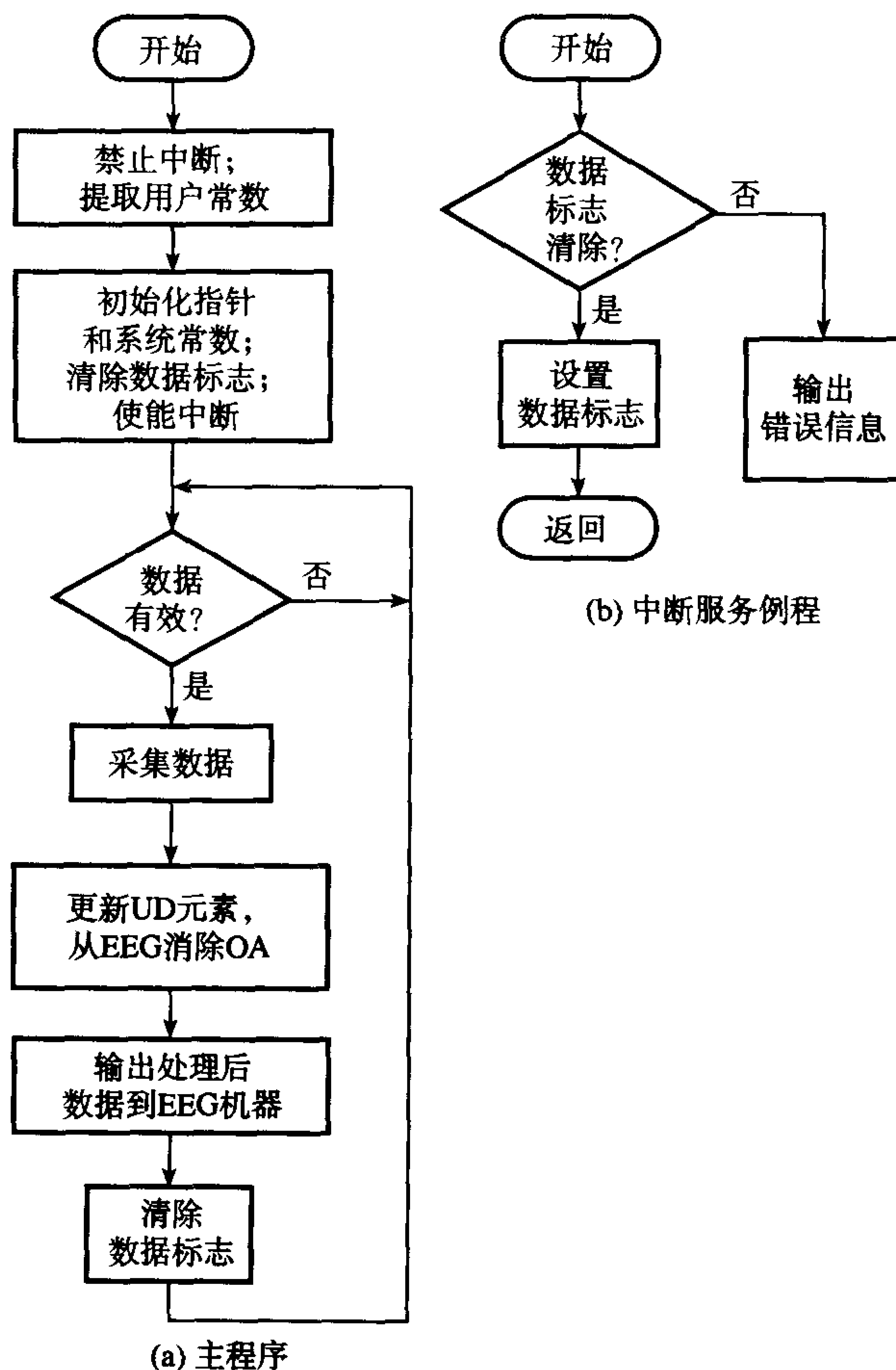
OAR 系统是中断驱动的。中断信号来自系统控制器板上的可编程时钟模块（PTM），具有 128 Hz的频率（有时使用95 Hz的频率以允许更多的计算时间，频率的改变很容易用软件实现）。发生中断后，OAR 系统软件用来采集EOG/EEG数据，使用UD算法从EEG样本消除OA，输出修正的EEG 和/或原始数据到EEG机器，这样就产生一张图表记录。系统操作总结在图 14.14 中。

在初始化阶段，通过一个可视的显示单元（VDU）邀请用户规定各种系统常数，即要修正伪像的EEG通道个数、模型参数个数以及在消除算法中应该使用的模型，还有输出到EEG机器的修正的EEG和/或原始EEG信号的个数。一些EOG信号和参数估计也可能输出到EEG机器。检查这些参数，如果有效，就用于初始化系统。对于任何无效的参数使用默认值。初始化之后，程序无限循环直到有效的数据可用（参见图 14.14(a)）。这里描述的过程只应用于原型 OAR 系统。在未来的OAR系统中，用户不再需要VDU，任何选择都通过按钮进行，“后台”程序将被更有用的自行操作例程替换（参见 14.2.2.3 节）。每次中断发生后，中断服务例程（参见图 14.14(b)）中的一个标志（数据位）就将置位，这表示主程序有效数据现在可用。数据采集之后，更新UD算法中的元素从EEG消除OA，输出修正的EEG和/或原始数据到EEG机器。最后，清除数据标志，指示当前数据样本已经成功处理。

如果在前一个中断服务结束前发生中断，就会向VDU输出一个错误信息，程序停止。这种情况一般发生在如果规定的参数和/或EEG通道超过了软件在8 ms的抽样间隔内的处理能力的时候，并防止不能服务的中断的积累和最终的系统崩溃。

OAR 系统中的算术操作使用浮点格式执行，以利用它提供的数值动态范围增加的优点，从而避免与使用定点方法相关的缩放问题。

因为速度在此应用中是至关重要的，硬件浮点被认为是最好的方法，但是发现当时可用的硬件浮点器件既昂贵又太慢。因此，降低了要修正的EEG通道的个数，这样就可以使用软件FP直到快速的 FP 器件可用。



(a) 主程序

(b) 中断服务例程

图 14.14 视觉伪像消除系统软件

14.2.2.5 系统测试和实验结果

在Plymouth的Freedom Fields医院进行了OAR系统的一项临床前测试。在第一个阶段，使用六名接受实验者进行了OAR系统可靠性的充分测试。这个阶段也是一个“学习”阶段，理解系统的行为和识别小的错误。在第二个阶段，一名非协作的接受实验者，他的EEG包含尖刺（spike）和波形放电（wave discharge）；还有两名接受实验者，其中一个是非协作的，他的EEG包含慢波，用来评估OA消除处理如何影响尖刺和慢波。在所有进行的测试中，要求接受实验者进行眼部运动练习，包括重复性的和随机的眨眼，以及垂直眼运动（VEM）和水平眼运动（HEM）。

从几个如图14.15所示放置的电极引出九个EEG信号。这些信号是FP2-F4、F4-C4、C4-P4、FP1-F7、F3-C3、C3-P3、Fz-Cz、Cz（以右耳垂或A2为参考）和Cz-Pz。EOG信号从放置在眼睛附近的电极引出，如图14.15(b)所示。

EEG和EOG信号通过头盔馈入一个八通道的EEG机器，放大后通过一个37线的D型接头馈入OAR系统。从EEG信号消除OA后，修正的和原始的两个EEG（消除均值）和/或EOG馈入EEG机器的后放大器，其后到纸张图表用于检查（参见图14.15(a)）。

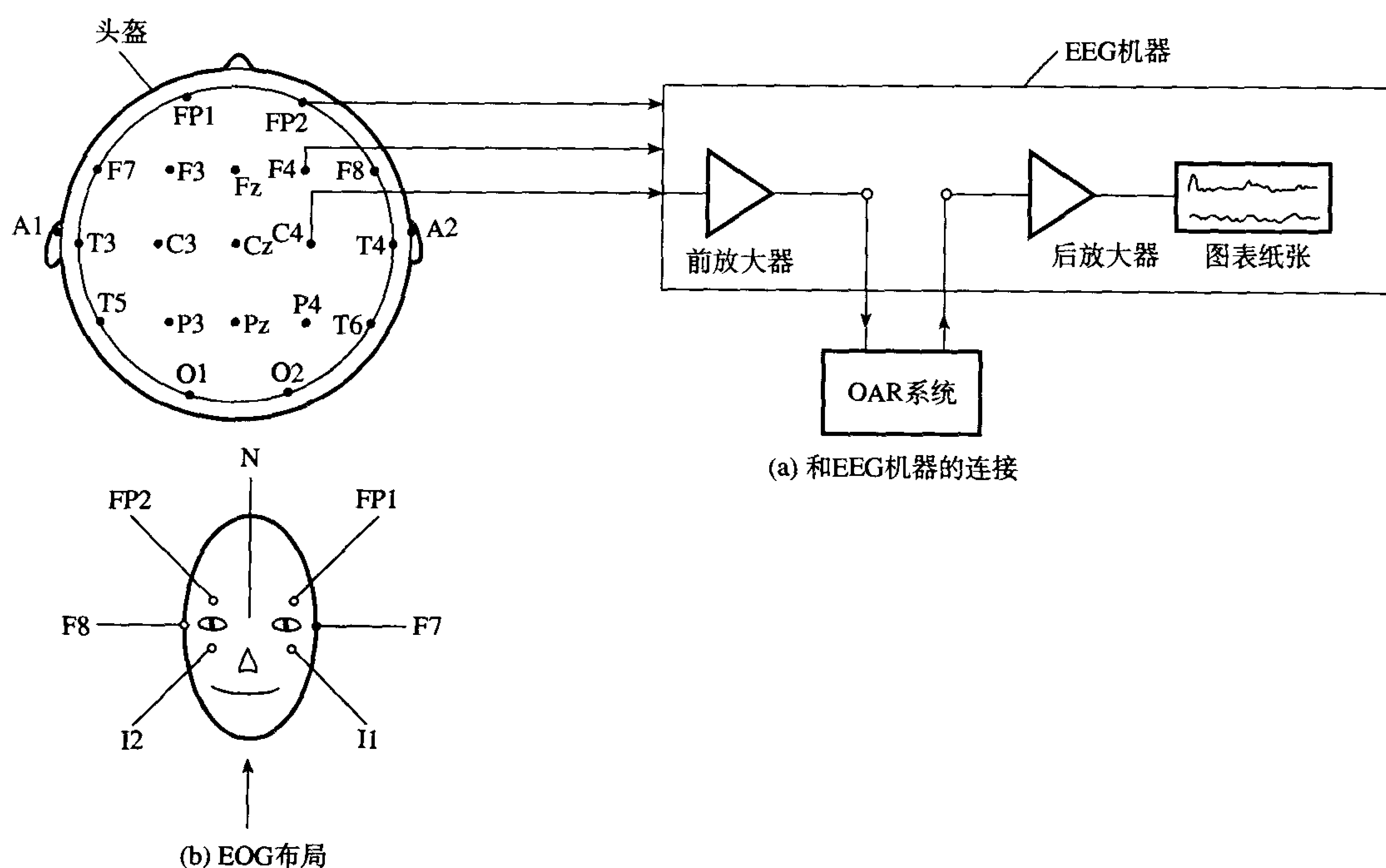
在测试中使用了几个模型，但是只有三个给出最好结果的模型将在这里描述。其中两个使用了从图14.15(b)中放置的电极引出的EOG信号，发现在前面的研究中也给出了最好的OA消除。这两个模型如下：

$$
\begin{aligned}
&\text{3D} \quad y(i)=\theta_1\mathrm{VR}(i)+\theta_2\mathrm{HR}(i)+\theta_3\mathrm{HL}(i)+e(i) \\
&\text{4D} \quad y(i)=\theta_1\mathrm{VR}(i)+\theta_2\mathrm{HR}(i)+\theta_3\mathrm{HL}(i)+\theta_4\mathrm{HL}(i)\mathrm{HR}(i)+e(i)
\end{aligned} \tag{14.16}
$$

第三个模型，为了符合前面的术语将其称为模型2H，使用了从FP1-F7和FP2-F8两对电极中引出的EOG：

$$
\text{2H} \quad y(i)=\theta_1\mathrm{EOGR}(i)+\theta_2\mathrm{EOGL}(i)+e(i) \tag{14.17}
$$

还应该提到EEG机器上的选择开关可以用于“强迫”OAR系统实现各种模型，通过选择合适的EOG电极对来馈入通道 1 到 4（这是为 EOG 保留的）。



(a) 和EEG机器的连接

(b) EOG布局

图 14.15　使用的实验排列

使用模型 2H（偶尔使用模型 3D 和 4D），OAR 系统可以用于从很多其他 EEG 电极中消除 OA（参见图 14.15(a)）。发现在所有研究的例子中，OA 都令人满意地消除了，这包括所有的前端 EEG 通道，那里的 OA 是最大的。图 14.16 给出了眨眼实验的四个不同电极（Fz-Cz、Cz-A2、F4-C4 和 F3-C3）的结果。在图 14.16(a)和图 14.16(b)中，同时修正了两个不同 EEG 信号的 OA。在两组图中修正的和未修正的EEG信号的比较说明系统令人满意地消除了OA（在每组图中将迹线(v)和(vi)与迹线(iii)和(iv)进行比较）。

更靠后放置的EEG电极，例如Cz-Pz和C4-P4几乎没有OA污染。在这些情况下，所有模型都执行得同样好。

图 14.17 显示了一段从一个非协作接受实验者得到的、包含癫痫尖刺和波形放电以及 OA 的 EEG 记录。（在本例和所有其他 EEG 包含异常波形的例子中，原始 EEG 如前面描述的那样直接馈给 EEG 机器的后放大器以及 OAR 系统，以允许对记录的无偏分析。）修正的 EEG 和原始的 EEG 的比较（参见图 14.17(d)和图 14.17(e)）说明 OA 已经消除，而没有处理尖刺和波形。

从一个非协作的精神病患者的EEG分析中（其EEG只包含小幅度的慢波）也得到了好的结果。值得注意的是，前面的在线方法不适合于协作接受实验者。

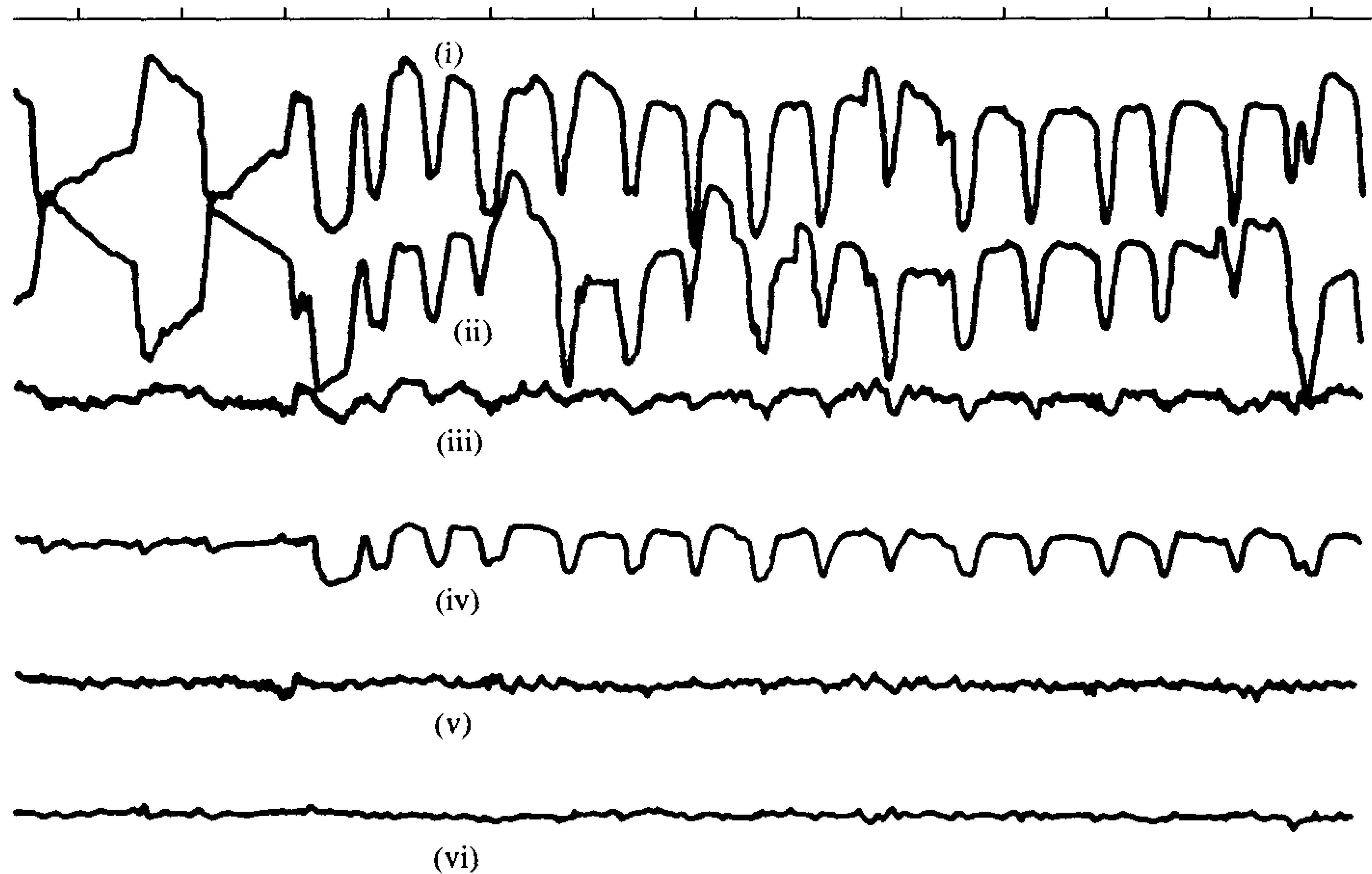


图 14.16 在一次眨眼实验中同时自适应消除一对EEG电极的视觉伪像。(i)和(ii)针对右眼和左眼测量的EOG信号；(iii)和(iv)在Cz-A2和Fz-Cz电极测量的EEG；(v)和(vi)消除了伪像的对应EEG

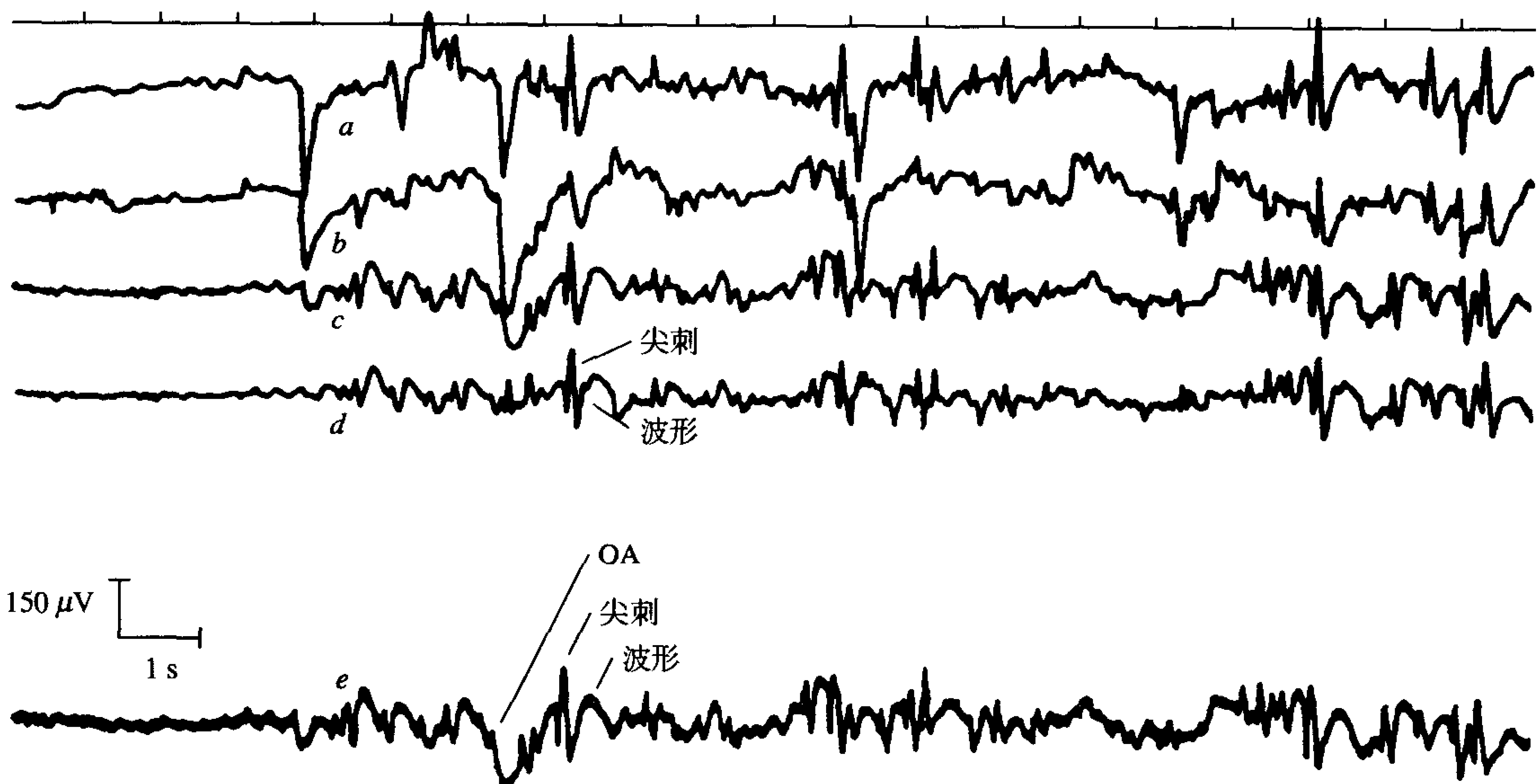


图 14.17 在出现癫痫尖刺和波形放电情况下的视觉伪像消除。(a)和(b)针对右眼和左眼测量的 EOG；(c) 测量的 EEG；(d) 消除伪像的 EEG；(e) 原始 EEG

14.2.2.6 讨论

使用不同类型的眼部运动（眨眼、VEM、HEM）的测试表明，使用数值稳定的UD算法，在所有EEG位置，满意地消除由于这些原因造成的OA是可能的。发现在更靠后的EEG位置，观察到OA污染非常少的EEG，在这些情况下所有模型执行地都很好。同时还发现，尽管在垂直眼运动时可以得到满意的OA消除，但是由于浮动伪像（rider artefact）造成的OA没有完全消除。在前面的研究中得到了相似的结果，这个结果确认了那些结果。OAR系统不能完全消除由于浮动伪像造成的OA，可能是因为使用的模型只考虑了EOG和EEG中的同时变化。形式为

$$y(m) = \sum_{n=0}^{N-1} h(n)x(m-n) + e(n)$$

的动态模型可以用于在希望得到改善结果的地方。

病人接受实验者的结果表明，当病理慢波和尖刺在没有OA的时候出现时，它们一般不会受OA消除处理的显著影响。然而，当它们和OA同时出现时，它们的幅度可能会减小但是不会完全消除。幅度减小主要发生在前额EEG通道。因此有必要在OA和慢波之间进行识别。这是另一个需要进一步研究的领域。

当前OAR系统的限制在于，甚至使用最简单的模型（模型2H），它也只能同时消除最多四个EEG信号中的OA，这是因为浮点算术例程的速度很低，其通常花费 70 μs 执行一次算术操作。这个问题的一个解决办法是使用快速硬件浮点算术器件，能够在 1 μs 或 2 μs 内执行一次算术操作。

14.2.2.7　结论

使用正常人和病人两种接受实验者得到的初步结果表明，对于眨眼、垂直和水平的眼运动以及双极的EEG电极安装，OAR系统给出了令人满意的OA消除。使用UD因子分解算法和软件控制的系统使我们能够克服前面在线OA消除方法的缺点。因此OAR系统能够处理多个伪像，在预备校准时不需要接受实验者的协作，消除准则基于完全客观的方法。这个系统，是它这一类中的第一个，与标准EEG机器兼容，所以可以作为一个附件进行生产和销售。然而，该仪器的用途只有经过充分的临床试验才能全面评估。

尽管OAR系统是专门为消除EEG中的OA而设计的，它也可以作为一个通用的伪像（或噪声）消除系统，在大多数污染和被污染信号能够分别测量的生理学场合使用。一个例子是在大的母体污染ECG出现的情况下测量胎儿ECG的问题（Widrow et al., 1975）。另一个例子是有必要从EEG中消除OA和OAR两个伪像的情况（Fortgens and De Bruin, 1983）。在这两种应用中，OAR系统经过软件和硬件可能的小修改之后，可以用于消除伪像。OAR系统，经过适当的编程，也可以用于其他的信号处理应用中，例如数字滤波。

自从大约10年前开始设计OAR系统以来，DSP特别是DSP硬件领域已经发生了相当大的变化。如果现在实现这个系统，有可能使用一个好的DSP芯片。今天，浮点数字信号处理器（比如TMS320C30或TMS320C40）对于时间紧要的系统（比如本系统）是适用的。即使DSP发生了变化，设计原理和问题仍旧不变。我们已经强调了这么多问题，希望读者将来可以从我们的经验中受益。

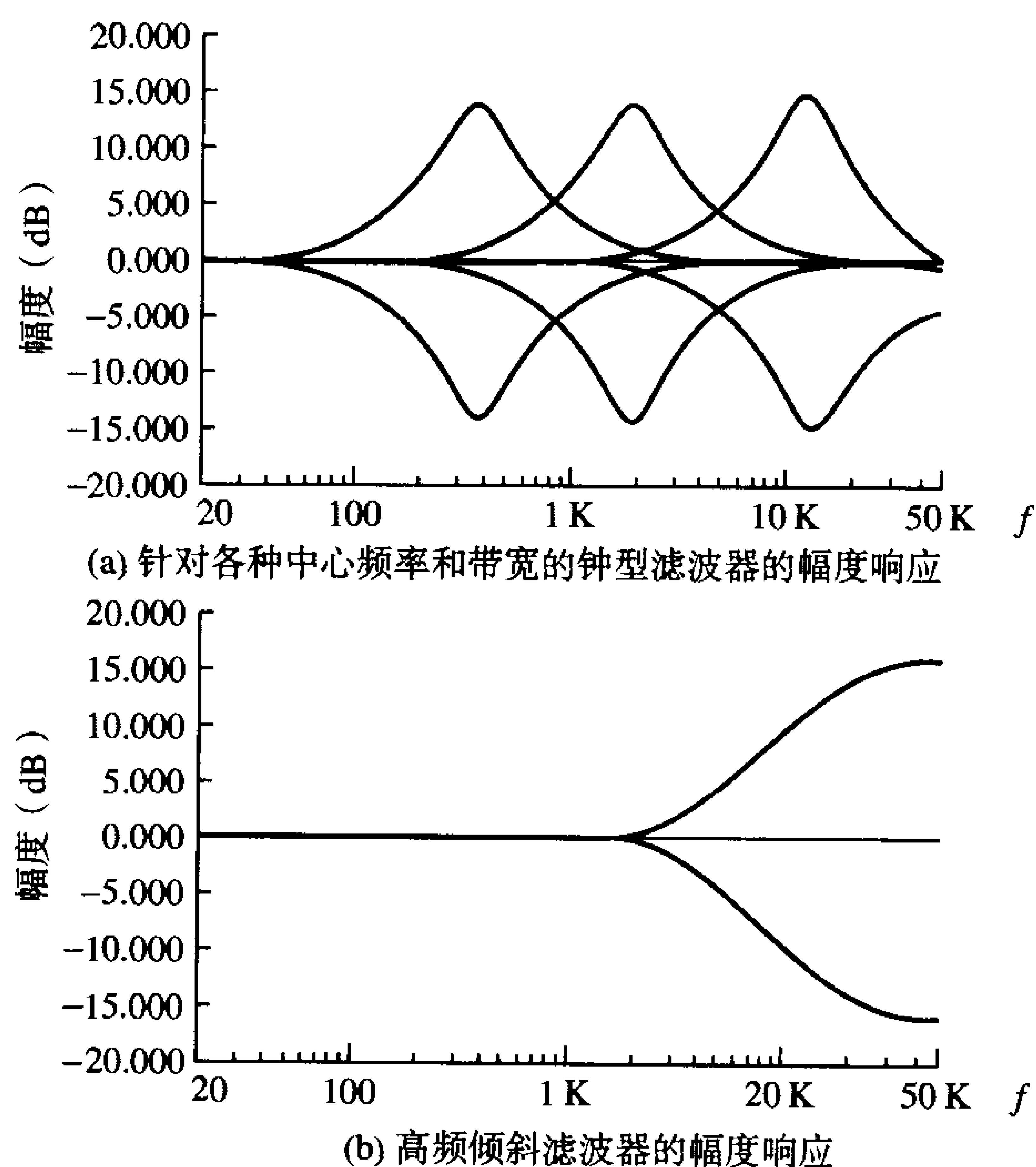
14.2.3　数字音频信号的均衡①

音频信号的均衡是很多专业和半专业音频应用（例如播音室录音、公共演说系统中的声音加强以及广播中使用的混音控制台）的一项重要的功能要求。音频均衡器基本上是一组具有可以调节的频率响应的滤波器，可以用某种期望的方式来对音频信号的频谱进行整形。在传统的混音控制台中，音频信号的均衡是通过模拟滤波实现的，但是趋势是朝着全数字混音控制台发展，因为其提供了改善的声音质量和未来生产成本的潜在减少。在全数字混音器中，模拟滤波器将由等效的实时数字滤波器替换。我们这里描述一个针对音频信号的实时半参数化均衡器，使用一款高速浮点数字信号处理器 TMS320C30 实现。

标准参数化均衡器允许用户在音频带内扫描特定的频率，在单频或一定频率范围内调节音频信号的电平。这里使用了三种基本滤波器类型。

① 本节的内容是基于 Robin Clark 的项目编写的。

- **钟型（bell）滤波器** 这种滤波器允许用户在音频带内放大或衰减特定的频率。钟型滤波器基本上是具有可调节的增益、Q因子和中心频率的带通滤波器。中心频率可以在20 Hz ~ 16 kHz的范围内变化，Q值在0.5 ~ 3之间，增益在±15 dB区间内。图14.18(a)举例说明了钟型特征。
- **倾斜（shelf）滤波器** 这种滤波器允许在音频带低端或高端的一定频率范围内调节均衡器的增益和截止频率。低频倾斜滤波器用于放大或衰减低频带，例如在20 ~ 500 Hz之间，而高频倾斜滤波器用于放大或衰减高频带，例如在1.6 ~ 16 kHz之间。家庭hi-fi系统中熟悉的高音或低音控制基本上是固定响应的倾斜滤波器。图14.18(b)举例说明了倾斜滤波器的典型响应。
- **通过（pass）滤波器** 这些滤波器基本上是带有固定截止频率的低通和带通滤波器，用于从音频信号中消除低和/或高频噪声。



(a) 针对各种中心频率和带宽的钟型滤波器的幅度响应

(b) 高频倾斜滤波器的幅度响应

图14.18 钟形特征和倾斜滤波器的典型响应

全均衡是通过基本滤波器的组合效果实现的。在模拟均衡器中，滤波器的特征参数（增益、中心频率、Q因子等）由用户通过交互式控件（可变电阻）调节。在数字实现中，这种调节是通过实时改变与均衡器参数变化相对应的数字滤波器系数实现的。

典型的模拟参数均衡器的分析表明，上面描述的每种滤波器类型，可以看做带有下面形式的s平面传输函数的巴特沃斯滤波器：对于钟型滤波器（Clark et al., 2000），

$$H(s) = \frac{s^2 + A\omega_n s + \omega_n^2}{s^2 + B\omega_n s + \omega_n^2} \tag{14.18a}$$

对于低频倾斜滤波器，

$$H(s) = \frac{s^2 + 2A\omega_n s + A^2\omega_n^2}{s^2 + 2B\omega_n s + B^2\omega_n^2} \tag{14.18b}$$

以及对于高频倾斜滤波器，

$$H(s)=\frac{A^2s^2+2A\omega_n s+\omega_n^2}{B^2s^2+2B\omega_n s+\omega_n^2} \tag{14.18c}$$

这里

$$A=C/Q$$

$$B=1/Q$$

为了实现和传统模拟均衡器相似的性能，通过使用双线性z变换技术（参见第8章）变换上面给出的每个s平面传输函数，用等效数字滤波器替换上面的模拟滤波器。每个函数的z变换结果具有形式：

$$H(z)=\frac{az^2+bz+c}{dz^2+ez+f} \tag{14.19}$$

这里

$$a=P^2+AP+\omega_n^2,\ b=2\omega_n^2-2P^2,\ c=P^2-AP+\omega_n^2$$

$$d=P^2+BP+\omega_n^2,\ e=2\omega_n^2-2P^2, f=P^2-BP+\omega_n^2$$

$$P=\frac{\omega_n}{\omega_p},\quad \omega_p=\frac{2}{T}\tan\left(\frac{\omega_n T}{2}\right)$$

仿真研究表明，当均衡器参数在音频范围内调节时，应该使用浮点算术来迎合滤波器系数数值范围内的大变化。浮点算术在此应用中也是很吸引人的，它使通过BZT“实时”地重新计算滤波器系数更加简单，以允许均衡器特征参数的在线调节。

均衡器是在基于PC的TMS320C30评估模块上实现的，其中包含了一个TMS320C30处理器、一个14位ADC/DAC模块和一个软件开发包。基于TMS320C30的参数均衡器的简化框图在图14.19中描述。模拟音频信号（例如，从CD播放器）以大约18.9 kHz的速率数字化为14位，并传送给C30，在那里通过钟型、倾斜和/或通过滤波器进行数字滤波。键盘用于调节均衡器的参数（增益、频率、使用的滤波器类型）。VDU（视觉显示单元）动态地显示了均衡器的频率响应。

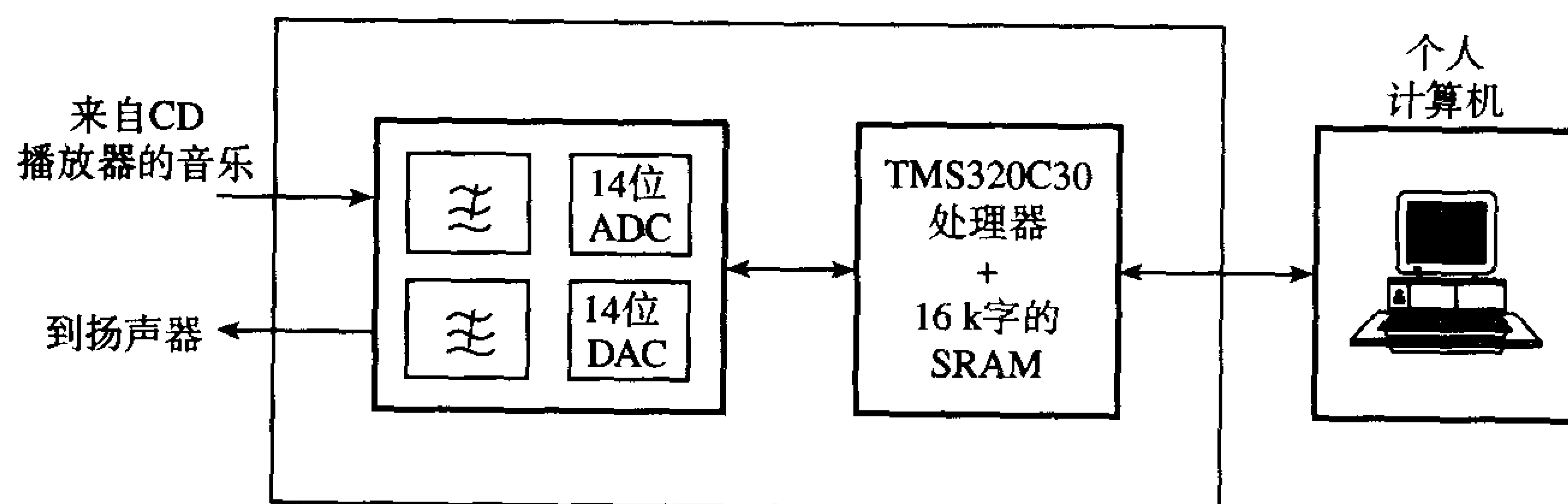


图 14.19　TMS320C30 EVM的简化框图

在EVM和PC中进行几个处理。C30处理器执行实现均衡所要求的滤波，当用户调节均衡器参数时为均衡器重新计算滤波器系数。系数的重新计算在后台进行，尽管滤波是中断驱动的。程序是用ANSI C语言编写的，使用浮点（24位尾数和8位指数）进行算术操作。滤波器和系数计算的C代码片断可以在指导手册中找到（细节请参见前言）。

使用CD播放器的音乐，针对各种参数设置评价了均衡器的性能，我们发现这种方式十分有效。当前的一个限制是EVM只允许最大大约18.9 kHz的抽样率，使得在这个阶段不能将该均衡器用在专业音频速率44.1 kHz。

14.3 设计学习

在大多数DSP课程中，设置一个或多个专业性的实际任务通常是有益的，这些任务是富有挑战性的，可以以小组或个人方式进行。这样的任务是想为学生提供机会，从而以直接的方式获得更深的DSP知识。其中用到了几个方面的DSP概念，可以在临近课程结束时或更早的时候设置（在后一种情况下，这些概念可以在它们正需要的时候进行教/学）。

在本节，我们介绍很多这样的专业问题，我们发现它们在一些课程中很有用。同时，本节还提供了学习目标和期望的子任务。

(1) 定点IIR数字滤波器舍入误差减小方案

图14.20显示了一个二阶IIR数字滤波器的结构，带有误差反馈方案以减小乘积舍入误差。该滤波器是在一个具有$B/2B$位架构的DSP处理器中，使用2的补码算术实现的。系数和其他变量以B位字存储，乘积在求和后量化。误差反馈系数每次只采用下列数值中的一个：0，±0.25，±0.5，±0.75，±1，±1.25，±1.5，±1.75，±2。

(a) 推导一般表达式：

- (i) 舍入噪声源$e(n)$和滤波器输出$y(n)$之间的传输函数；
- (ii) 舍入误差的输出噪声功率；
- (iii) 滤波器输出端的信噪比(SNR)，假定输入信号$x(n)$是已知频率的正弦波并且量化为B位；
- (iv) 反馈网络的零点位置。

(b) 开发：

- (i) 一个算法用于计算舍入误差的输出噪声功率、滤波器输出端的SNR和反馈网络的零点位置，给定未量化的滤波器系数、反馈系数（从可允许的设置中）以及系统字长B的数值；
- (ii) 一个实现(i)中算法的MATLAB程序。

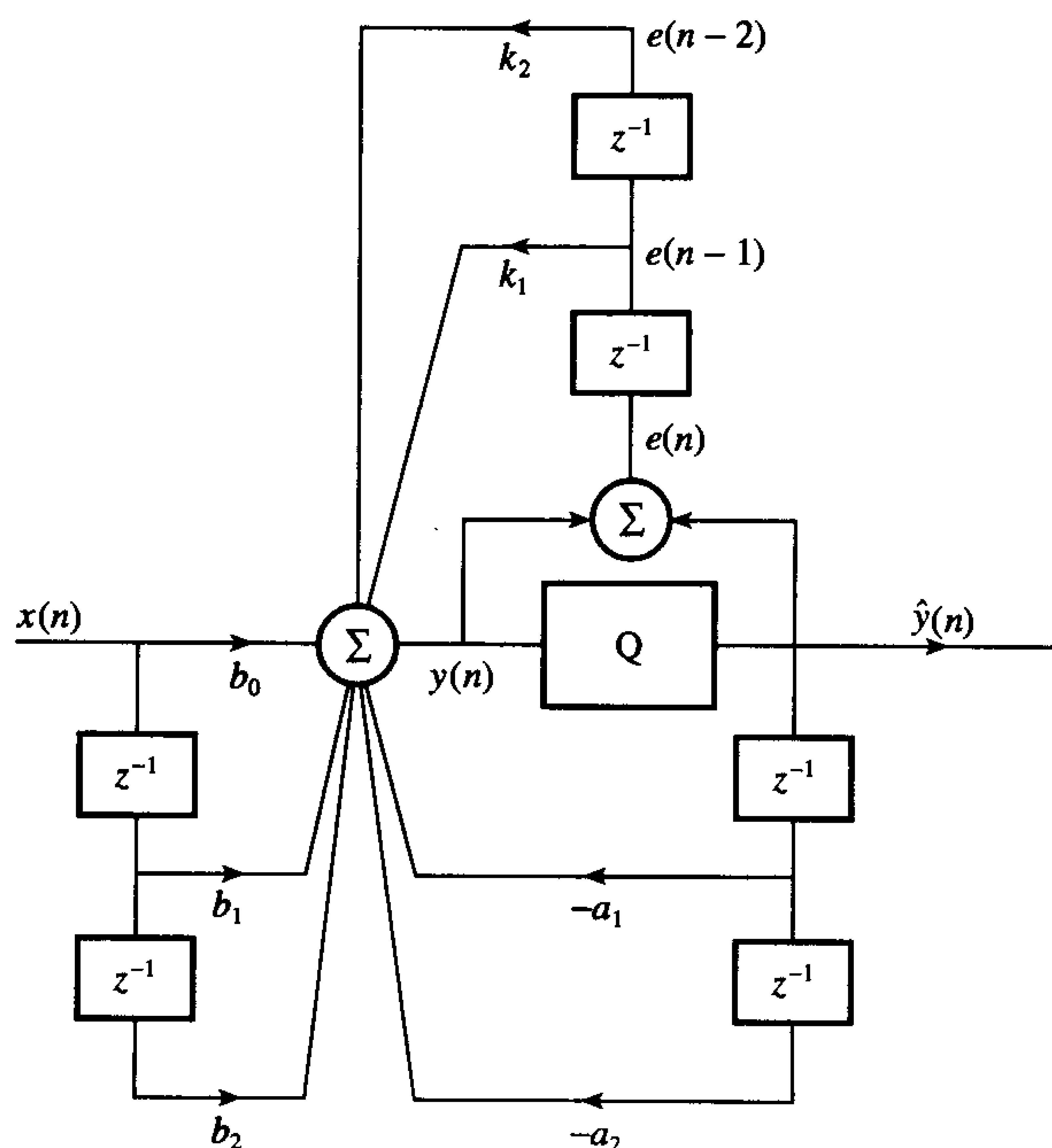


图14.20　二阶噪声减小方案

(c) 使用你的MATLAB程序以及MATLAB中的其他工具，对于下列每个滤波器，研究误差反馈方案对乘积舍入噪声电平的影响（假定 $B=16$ 位，然后对 $B=24$ 位重复进行）(Dattorro, 1983)：

(i) 一个用于音频信号处理的“剪切”（cut）滤波器，特征参数为

滤波器系数　$b_0=0.996\,450\,761\,790$，$b_1=-1.992\,821\,454\,486$，$b_2=0.996\,443\,656\,208$
$a_1=-1.992\,821\,454\,486$，$a_2=0.992\,894\,417\,998$

抽样频率　48 kHz

(ii) 一个用于音频信号处理的“放大”（boost）滤波器，特征参数为

滤波器系数　$b_0=0.996\,450\,761\,790$，$b_1=-1.992\,821\,454\,486$，$b_2=0.996\,443\,656\,208$
$a_1=1.992\,821\,454\,486$，$a_2=-0.992\,894\,417\,998$

抽样频率　48 kHz

中心频率　15 kHz

(iii) 一个高通滤波器，特征参数为

滤波器系数　$b_0=0.292\,893$，$b_1=-0.585\,786$，$b_2=0.292\,893$
$a_1=0$，$a_2=0.171\,572\,8$

抽样频率　8 kHz

阻带边沿频率　500 Hz

通带边沿频率　2 kHz

(d) 上面的研究应该包括：

(i) 对于每个滤波器和字长 B，针对表 14.2 中的每对反馈系数，输出噪声功率、SNR 以及反馈网络零点位置的计算和制表；

(ii) 确定给出最低输出噪声功率的误差反馈系数对（来自所有可能的系数对和在(a)部分中规定的可允许的反馈系数值）；

表 14.2　误差反馈系数值

序号	$k1'$	$k2'$
1	0	−1
2	0	1
3	−1	0
4	−1	−1
5	−1	1
6	1	0
7	1	−1
8	−2	−1
9	2	−1
10	0	0

(iii) 对于表 14.2 中的每对反馈系数和在(ii)发现的最好的系数对，画出对应于噪声传输函数的频率响应；

(iv) 写出一个关于此研究和你的发现的报告（最多 3 ~ 4 页，不包括图表、程序清单）。你的报告应该包括下面几项：

- DSP 系统中乘积舍入误差噪声问题的介绍；
- 你对上面(a) ~ (d)部分的详细回答；
- 你的结果的关键讨论以及在实际中如何选择误差反馈系数的建议；
- 作为一种减小舍入噪声的手段，误差反馈方案局限性的关键讨论；

- 误差反馈方案其他好处的讨论，舍入噪声减小除外；
- 对于自己已经从这项作业中学到的知识，做一个评价的总结；
- MATLAB 程序清单。

(2) 多速率滤波器设计学习

背景

这个设计学习的基础是基于 MATLAB 的滤波主题中的一个，是由我们以前的一个学生 Nick Outram 博士（Outram et al., 1995）为我们的 DSP 课程开发的。这里的学习目标包括

- 增强多速率信号处理的概念；
- 当期望的幅度响应非常尖锐时，FIR 滤波器一些局限性的证明；
- 提供设计和应用多速率滤波器的手把手（hand-on）的经验。

在这个设计学习中，我们将使用胎儿 ECG（心电图）以帮助学习。胎儿 ECG 的形状在临床上是很令人感兴趣的，但是怀疑它将出现称为基线偏移的低频伪像失真，请参见图 14.21。基线偏移妨碍胎儿 ECG 特征的准确测量和分析。困难在于信号增强不应该失真临床上感兴趣的低频信息。遗憾的是，大部分基线偏移能量低于 3 Hz。

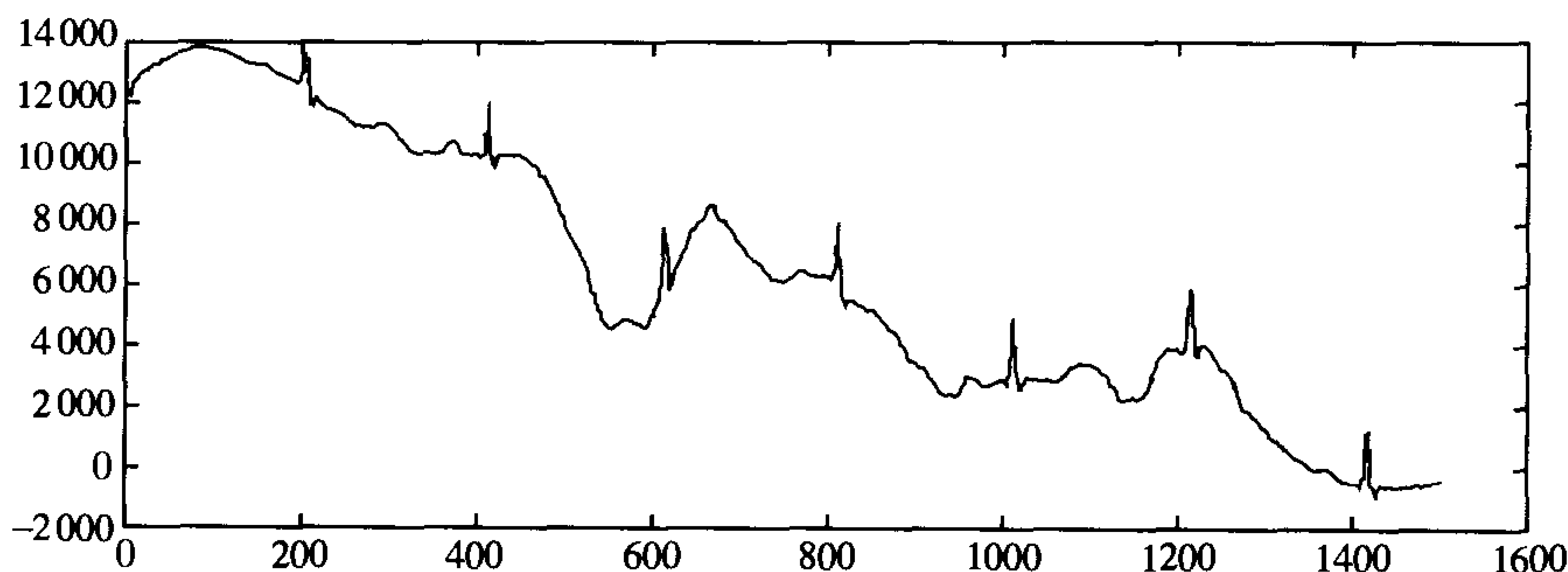


图 14.21　剧烈的基线偏移

在学习中，我们提供了两组数据：好的 ECG，包含高质量的胎儿 ECG 记录几乎没有基线偏移；以及差的 ECG，包含带有明显基线偏移的胎儿 ECG。胎儿 ECG 数据由以 500 Hz 抽样的 12 位的数据数值组成。

问题

主要问题是使用线性相位 FIR 滤波器，开发一个信号处理方案以消除基线偏移而且不会失真 ECG 形状。

规定任务是

(a) 编写一个 MATLAB 程序，将 ECG 数据文件读入两个独立的数组，并且在相同坐标轴上显示每个波形的前 4000 个样本；

(b) 使用 MATLAB 的 remezord 函数，估计滤波器系数的个数，要求满足图 14.22 中描述的指标。解释你的结果和设计窄带 FIR 滤波器的问题。

(c) 在 MATLAB 中使用合适的抽取和内插技术，设计和实现一个多速率滤波器，满足图 14.22 中的指标；

(d) 使用数据测试该滤波器，检查 ECG 波形的失真和确定通过多速率系统的延迟；

(e) 写出一个关于设计学习的报告，应该包括你的多速率滤波器的设计和测试。

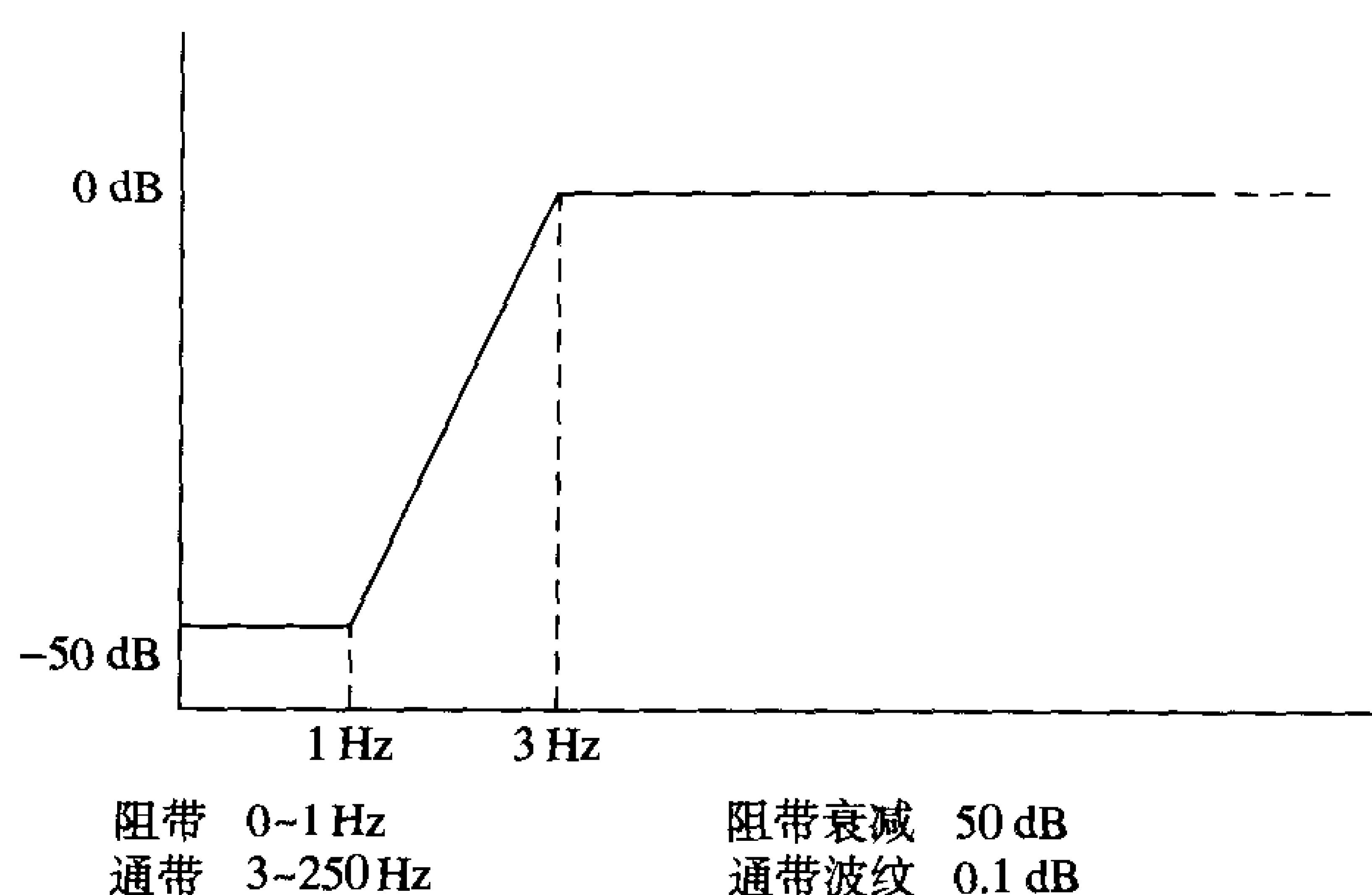


图 14.22 为了消除基线偏移的滤波器指标

(3) 使用 MATLAB 和 DSP56002 处理器设计和实现数字滤波器

背景

这个设计问题的目标是使学生能够获得使用MATLAB和定点DSP处理器设计和实现FIR滤波器的手把手的经验。这个设计问题的基础是Brahim Hamadicharef、Robin Clark 和 Eddie Riddington 为我们的一门 DSP 课程开发的实验室练习。

规定的学习目标是

(1) 使用 MATLAB

- 对于一组给定的指标，计算系数，画出 FIR 滤波器的频率响应；
- 研究系数量化对频率响应的影响；

(2) 使用定点 DSP 处理器实现 FIR 滤波器

- 熟悉 DSP 开发工具；
- 为一个定点 FIR 滤波器开发一个简单的汇编语言程序；
- 测试和验证实时滤波。

问题

对线性相位 FIR 滤波器的要求是从音频信号中消除 1 kHz 的干扰。滤波器应该满足下列指标：

通带波纹	0.5 dB
阻带衰减	25 dB
通带	900 ~ 1100 Hz
阻带边沿频率	990 Hz 和 1010 Hz
抽样频率	8 kHz

要求你使用MATLAB和摩托罗拉DSP56002定点DSP处理器来设计和实现该滤波器。规定任务如下：

(a) 使用MATLAB和最佳的方法计算一个合适的FIR滤波器的系数。注意。你应该使用MATLAB函数remezord 估计滤波器的长度N，使用函数remez计算滤波器系数，使用函数freqz画出滤波器的频率响应。下面的公式对于确定起伏参数可能很有用：

$$R_p = -20\log_{10}\frac{1-\delta_p}{1+\delta_p} \qquad \text{– 通带波纹 dB}$$

$$R_s = -20\log_{10}\frac{\delta_s}{1+\delta_s} \quad \text{– 阻带衰减 dB}$$

这里

δ_p – 通带波纹

δ_s – 阻带波纹

(b) 通过分析将系数量化为4位、8位、16位和24位对滤波器频率响应的影响，为滤波器确定一个合适的系数字长。注意。为了将滤波器系数量化成B位、2的补码的定点数，我们简单地将每个系数乘以2^{B-1}然后舍入到最近的整数。真实系数和定点表示二者之间的差给出了系数量化误差的一种度量。

(c) 在DSP56002评估模块上实现该滤波器，并且验证该滤波器的操作正确。注意，作为使用DSP56002的FIR数字滤波的入门知识，考虑图14.23中描述的简单三点FIR滤波器。该滤波器由下面方程表征：

$$y(n) = h(0)x(n) + h(1)x(n-1) + h(2)x(n-2)$$

图 14.23 一个简单的三点FIR滤波器

表14.3给出了滤波器系数和数据初始的存储器映像。寄存器r0和r4分别用做系数和数据存储器的指针，并且使用了循环寻址。程序14.1给出了实现这个简单滤波器的DSP56002的代码。注释列解释了程序14.1的操作。

表 14.3 初始存储器映像

X存储器		Y存储器	
$h(2)$	← r0	$x(n-2)$	← r4
$h(1)$		$x(n-1)$	
$h(0)$		$x(n)$	

程序 14.1 一个简单3点FIR滤波器的DSP56002代码

```
move  x: (r0)+, x0                       ; copy h(2) into register x0 and increment pointer r0.
move  y: (r4)+, y0                       ; copy x(n−1) into register y0 and increment pointer r4.
mpy   x0, y0, a x: (r0)+, x0 y: (r4)+, y0 ; multiply h(2) and x(n−2) and store in accumulator a.
                                         ; copy h(1) into register x0 and increment r0.
                                         ; copy x(n−1) into register y0 and increment r4.
mac   x0, y0, a x: (r0)+, x0 y: (r4)+, y0 ; multiply h(1) and x(n−1) and add to the accumulator a.
                                         ; Copy a0 into register x0 and increment r0.
                                         ; Copy x(n) into register y0 and increment r4.
mac   x0, y0, a y1, y: (r4)+             ; multiply h(0) and x(n) and add to the accumulator a.
                                         ; Accumulator a contains the output sample y(n).
                                         ; x(n−2), x(n−1) and x(n) shall now be referred to as
                                         ; x(n−3), x(n−2) and x(n−1). Overwrite x(n−3) with the
                                         ; next input sample stored in y(1).
```

对于具有很多系数的FIR滤波器，和mac指令一起使用DSP56002的重复指令将更有效率。DSP56002 代码的一个片断在下面给出：

```
; FIR filtering using the repeat instruction

clr     a           x(r0)+, x0    y(r4)+, y0
rep     #N-1
mac     x0, y0, a   x(r0)+, x0    y(r4)+, y0
macr    x0, y0, a   (r0)-
move    a, y:Output
```

(d) 编写一个报告，给出你的滤波器设计和实现的细节，包括MATLAB和DSP56002程序清单。

14.4 基于计算机的 DSP 多项选择题

在本节中，我们描述一种方法，我们发现这对于快速评价大群学生和识别需要采取补救行动的情况是有用的。这个方法的一个重要特征是它能够通过测试学生对确定问题的理解力而使他们修订DSP 材料。

本质上，我们开发了两套涵盖了基本的和高级的/应用的DSP主题的多项选择题库。题库每年都进行更新和扩大。关于基本DSP主题的题库用于在第一学期测试学生，高级的/应用的DSP题库则在第二学期测试他们。在每种情况下，大约在测试前两周，会给学生分发考试中可能会有的试题的硬拷贝，但是没有给出答案。我们不会给出试题的直接参考。软件Questionmark用来进行考试和自动打分。每次测试都是计算机计时，持续半个小时，从试题数据库中随机抽出大约30个试题。

所有试题或者是多项选择或者是多项回答，当前的评分方案如下（尽管这可以很容易修改）：

多项选择	正确答案	+1
	错误答案	−1
	没有答案	0
多项回答	正确答案	+1
	错误答案	−1
	没有答案	0

多项选择题的例子在下面给出。有关多项选择题的细节，包括当前试题数据库的清单，可以在指导手册中找到（细节请参见前言）。

(1) 关于基本 DSP 主题的样题

试题 1

一个要进行频谱分析的模拟信号以125 Hz抽样产生1024个数据值。频谱样本之间的频率间隔是多少？

(a) 0.041 42 Hz

(b) 0.122 07 Hz

(c) 0.000 98 Hz

(d) 0.0080 Hz

试题 2

一个数字滤波器是一个 IIR 如果

(a) 它的所有极点位于单位圆之外

(b) 一个或更多分母系数不为零
(c) 当前输出依赖于以前的输出
(d) 它出现振荡

试题 3

一个离散时间滤波器传输函数 $H(z)$为

$$H(z) = 1/(1 + 0.454\,456z^{-1} + 0.269\,259z^{-2})$$

找出滤波器的极点：
(a) 极点是：0.45+0.75j 和 0.45–0.75j
(b) 极点是：–0.2272+0.4664j 和 –0.2272–0.4664j
(c) 极点是：–0.9452+0.5j 和 –0.9452–0.5j

试题 4

带通滤波器可以通过将合适的原型低通模拟滤波器变换为带通滤波器设计。这种变换
(a) 将模拟低通滤波器的阶数减半
(b) 将模拟低通滤波器的阶数加倍
(c) 将低通滤波器的截止频率 F_c 和 $-F_c$ 映射为带通滤波器的截止频率 F_{c2} 和 F_{c1}
(d) 将低通滤波器的截止频率 F_c 和 $-F_c$ 映射为带通滤波器的中心频率

试题 5

一个离散滤波器的极点位于 $z = \mathrm{j}0.75$ 和 $z = -\mathrm{j}0.75$，零点位于 $z = -1$ 和 $z = 1$。抽样频率四分之一处的幅度响应是多少？
(a) 1.52
(b) 4.57
(c) 7.15

试题 6

将系数 -0.1743 量化为 8 位得到
(a) –22
(b) –23
(c) +22
(d) 23×10^{-7}

试题 7

下列哪个应用在 DSP 中？
(a) 奈奎斯特频率等于抽样频率的一半
(b) 奈奎斯特频率等于抽样频率
(c) 奈奎斯特频率等于折叠频率
(d) 奈奎斯特频率是过抽样率的一半

试题 8

一个离散时间滤波器的传输函数为

$$H(z) = (1 - 1.6z^{-1} + z^{-2})/(1 - 1.5z^{-1} + 0.8z^{-2})$$

直流处的幅度响应是多少？

(a) 0

(b) 1.33

(c) 1

(d) 1.6

试题 9

一个离散时间滤波器传输函数为

$$H(z) = (1 - 1.6z^{-1} + z^{-2})/(1 - 1.5z^{-1} + 0.8z^{-2})$$

极点和零点的半径是多少？

(a) 0.9, 1

(b) 0.81, 1

(c) 1, 0.81

(d) 1, 1

(e) 0.5, 0.5

试题 10

关于 IIR 滤波器设计的双线性 z 变换（BZT），符合下面的哪一项？

(a) 越靠近奈奎斯特频率 BZT 方法的畸变（warping）效应越差

(b) 在整个频率响应对 BZT 引起的失真进行预畸变（pre-warping）补偿

(c) 只在特定的频率对 BZT 引起的失真进行预畸变补偿

(d) 频率缩放在频率响应中消除了畸变效应

试题 11

对于具有下列指标的高通 IIR 滤波器有一个要求：

阻带边沿频率	1 kHz
通带边沿频率	2 kHz
抽样频率	16 kHz
通带波纹	3 dB
阻带衰减	30 dB

对于一个合适的原型低通滤波器，通带和阻带频率是多少？假定采用双线性 z 变换设计方法。

(a) 0.0414, 0.0198

(b) 1, 2

(c) 1, 2.0813

(d) 2, 1

(e) 0.0198, 0.0414

试题 12

一个 DSP 系统的模拟输入信号由一个截止频率为 10 kHz 的四阶巴特沃斯滤波器限带，然后以 50 kHz 抽样。假定一个宽带输入信号。10 kHz 处的混叠误差电平是多少？

(a) 0.156

(b) 0.026

(c) 0.04

(d) 0.707

试题 13

一个DSP系统的模拟输入信号由一个截止频率为10 kHz的四阶巴特沃斯滤波器限带，然后进行抽样。假定一个宽带输入信号。确定在10 kHz处得到20 dB信号混叠误差电平的最小抽样频率。

(a) 10 kHz

(b) 20 kHz

(c) 29.39 kHz

(d) 19.39 kHz

(e) 50 kHz

试题 14

一个音频信号具有从0扩展到24 kHz的频率成分。信号以3.072 MHz的速率抽样。过抽样比是多少？

(a) 32

(b) 64

(c) 128

(d) 256

试题 15

一个模拟信号根据带通抽样理论抽样。假定信号占据48 ~ 52 kHz的频带，为了避免混叠，最小的理论抽样速率是多少？

(a) 104 kHz

(b) 96 kHz

(c) 8 kHz

(d) 16 kHz

试题 16

一个由32个样本组成的数据序列以自然顺序保存在存储器中。如果数据序列是严重搅乱的，使用标准倒位算法，数据样本$x(7)$和$x(13)$的倒位下标是多少？假定使用32点的FFT处理器。

(a) $x(11)$和$x(14)$

(b) $x(22)$和$x(28)$

(c) $x(7)$和$x(13)$

(d) 以上都不是

试题 17

对于一个3点DFT，给出下列4点DFT系数：

(a) $X(0) = 1.5$

(b) $X(1) = 1-0.5j$

(c) $X(2) = 0.5$

(d) $X(3) = 1+0.5j$

等效的离散时间序列是什么？

(a) $x(0) = 1, x(1) = 0, x(2) = 1, x(3) = 0$

(b) $x(0) = 0.5, x(1) = 1, x(2) = 0, x(3) = 0$

(c) $x(0) = 1, x(1) = 0.5, x(2) = 0, x(3) = 0$

(d) $x(0) = 0.5, x(1) = 0.5, x(2) = 0, x(3) = 0.5$

(e)以上都不是

试题 18

一个 8 点基 -2 FFT 的前四个输出是

$X(0) = 27$

$X(1) = -4+3j$

$X(2) = 4+j$

$X(3) = 0-5j$

下列陈述中哪一个是正确的?

(a) $X(7) = 0+5j$

(b) $X(7) = -4-j$

(c) $X(7) = -4-3j$

(d) 输出序列的直流值是 27

(e) 以上都不是

(2) 关于高级 / 应用 DSP 主题的样题

试题 1

下列陈述中哪一个是正确的？在数字通信中，边界稳定的 IIR 滤波器用于时钟恢复（clock recovery），因为

(a) 它们的极点和零点靠近单位圆

(b) 它们具有随时间缓慢衰减的冲激响应

(c) 这确保了有一个时钟信号，甚至是在相当长的周期内输入数据流中没有变化时

(d) 它们不易随时间和温度漂移

(e) 它们适用于涉及到突发传输的应用

试题 2

一个数字通信系统具有56 k波特的数据率和每秒448 k个样本的抽样率。该系统的时钟恢复IIR滤波器具有 25 Hz 的带宽。对于该滤波器，下列哪一项是正确的?

(a) 极点位置是: $r = 0.9998$，±45 度

(b) 极点位置是: $r = 0.9998$，45度

(c) 极点位置是: $r = 0.9998$，±0.785 度

(d) 和极点相关的系数是: $a_1 = -1.413\ 965, a_2 = 0.999\ 825$

(e) 和极点相关的系数是: $a_1 = -1.413\ 965, a_2 = 0.999\ 649$

(f) 时钟恢复滤波器是一个带通滤波器

(g) 时钟恢复滤波器是一个低通滤波器

试题 3

一个数字通信系统具有 28 k 波特的数据率和每秒 224 k 个样本的抽样率。时钟恢复 IIR 滤波器具有 100 Hz 的带宽，分母系数 $a_1 = -1.412\ 230$，$a_2 = 0.997\ 196$。下列陈述哪个是正确的，如果滤波器系数量化为 8 位定点数?

(a) 定点系数是: $a_1' = -90, a_2' = 63$

(b) 定点系数是: $a_1' = -181, a_2' = 127$

(c) 用小数表示，量化的系数是：$a_1' = -1.406\ 25, a_2' = 0.984\ 375$

(d) 用小数表示，量化的系数是：$a_1' = -1.414\ 062, a_2' = 0.992\ 187$

(e) 以上都不是

试题 4

关于使用二阶标准节实现的定点 IIR 滤波器，识别正确的陈述：

(a) 溢出错误是由超出可允许字长的加法造成的

(b) 溢出能引起自持续（self-sustaining）振荡

(c) 一般来说，频域缩放（切比雪夫范式）完全消除了溢出误差

(d) 误差反馈或噪声整形技术可以用于减小溢出误差

(e) 溢出的缩放可以改善滤波器输出端的信噪比

试题 5

一个 IIR 数字陷波滤波器，陷波频率是抽样频率的四分之一，具有下列系数：$b_0 = 1, b_1 = 0, b_2 = 1; a_1 = 0, a_2 = 0.81$。在滤波器的输入端频域缩放因子是多少？假定使用二阶标准节实现。

(a) 1.81

(b) 0.55

(c) 5.26

(d) 0.19

(e) 以上都不是

试题 6

下列陈述中哪一个是正确的？关于 DTMF 解码，对每个 Goertzel 滤波器只要求一个实系数，因为

(a) 只有音调频率的幅度要求解码 DTMF 信号

(b) 在 DTMF 解码中不要求相位信息

(c) Goertzel 滤波器在每个样本到来时处理，不像在 FFT 中等待一组 N 个样本

(d) 适合于 DTMF 解码的 Goertzel 滤波器执行速度快，占用存储空间少

(e) Goertzel 算法由一系列二阶 IIR 滤波器组成

试题 7

一个用于数字电话的 DTMF 音调检测方案使用了一系列二阶 Goertzel 滤波器提取 DTMF 音调和它们的谐波。假定数码‘0’的 DTMF 音调是 941 Hz 和 1336 Hz。确定用于低频音调 941 Hz 的两个 Goertzel 滤波器反馈回路的系数值，如果对于基频和二次谐波，对应的离散频线是 24 和 27，分别使用 $N = 205$ 和 $N = 210$ 的序列长度。

(a) 对低频音调例如 941 Hz，反馈系数值是 $a_1 = 0.999\ 71, a_2 = -1$

(b) 对低频音调例如 941 Hz，反馈系数值是 $a_1 = 1.482\ 867, a_2 = -1$

(c) 对低频音调例如 941 Hz，反馈系数值是 $a_1 = 1.345\ 621, a_2 = -1$

(d) 对于基频音调的二次谐波例如 2×941 Hz，反馈系数值是 $a_1 = 0.463\ 812, a_2 = -1$

(e) 对于基频音调的二次谐波例如 2×941 Hz，反馈系数值是 $a_1 = 0.488\ 851, a_2 = -1$

(f) 对于基频音调的二次谐波例如 2×941 Hz，反馈系数值是 $a_1 = 0.327\ 635, a_2 = -1$

试题 8

下列陈述中哪一个是正确的？在 DTMF 解码中：

(a) 我们需要知道 DTMF 频率的二次谐波以及基频的幅度以区分语音和 DTMF 音调

(b) 我们不需要知道 DTMF 频率二次谐波的幅度就可以区分语音和 DTMF 音调

(c) 电话系统的频率响应是这样的，DTMF 高频音调比低频音调衰减要大

(d) 电话系统的频率响应是这样的，DTMF 低频音调比高频音调衰减要大

(e) 语音具有显著的偶次谐波，而 DTMF 信号没有

(f) DTMF 信号具有显著的偶次谐波，而语音没有

试题 9

在一个系统中使用一个二级抽取器将抽样频率从每秒 240 k 个样本减小到每秒 8 k 个样本。每一级的抽取因子是 $M_1 = 15$ 和 $M_2 = 2$，滤波器长度是 $N_1 = 45$ 和 $N_2 = 43$。下列陈述中哪一个是正确的？

(a) 两级输出的抽样率分别是每秒 16 k 和 8 k 个样本

(b) 两级输出的抽样率分别是每秒 8 k 和 16 k 个样本

(c) 抽取器复杂度的度量是：MPS = 1 064 000, TSR = 88

(d) 抽取器复杂度的度量是：MPS = 1064, TSR = 88

(e) 以上都不是

试题 10

使用一个二级抽取器将抽样率从 500 Hz 减小到 12.5 Hz。每一级的抽取因子是 10 和 4，相关的滤波器长度分别是 $N_1 = 55$ 和 $N_2 = 97$。抽取器的效率指标是

(a) MPS = 32 350.5, TSR = 152

(b) MPS = 3962.5, TSR = 152

(c) MPS = 500, TSR = 55

(d) MPS = 28 712.5, TSR = 152

(e) 以上都不是

试题 11

使用一个二级抽取器将抽样率从 500 Hz 减小到 12.5 Hz。第一级和第二级抽取的抽取因子分别是 10 和 4。感兴趣的频带是 0 ~ 4 Hz。在第一级中的防混叠滤波器的带沿频率是

(a) 0, 4, 6.25 Hz

(b) 0, 4, 43.75 Hz

(c) 0, 4, 12.5 Hz

(d) 0, 4, 50, 6.25 Hz

(e) 以上都不是

14.5　小结

在本章中，我们介绍了很多设计和开发板可以用于实现在本书中描述的一些 DSP 算法。本章以学习案例的形式描述了很多现实世界的DSP应用，从而给读者提供一些实际设计问题的思想。我们在习题部分还介绍了很多富有挑战性的设计学习，以及一种基于多项选择题的方法，这对于快速评价 DSP 主题是很有用的。两者都应该作为一种获得对 DSP 更深入认识的手段。

习题

14.1　说明一个被随机噪声污染的信号的自相关函数（ACF）就是信号本身的 ACF，说出所做的任何假定。解释这个结果如何用来检测隐藏的周期性。

14.2 证明数字匹配滤波器输出端的最大信噪比和输入信号的波形无关，说出所做的任何假定。

14.3 隐藏在噪声中的重复信号，可以通过数字匹配滤波器检测。下面给出的是无噪声信号和含噪声信号的连续样本值：

无噪声信号 {−0.51, −0.35, −0.29, −0.25, −0.29, −0.39, −0.47}

含噪声信号 {−0.18, −0.06, 0.27, 0.69, −0.50, −0.44, −0.20, −1.46, −0.93, −1.46, −0.91, −0.39, −1.70}

确定

(a) 数字匹配滤波器的系数；

(b) 数字匹配滤波器的输出；

(c) 匹配滤波获得的信噪比改善，用分贝表示。

注意：滤波器输出端噪声的方差 σ_0^2 为

$$\sigma_0^2 = \sigma^2 \sum_{m=0}^{\infty} h^2(m)$$

这里 σ^2 是滤波器输出端噪声的方差，$\{h(m)\}$是滤波器系数。

参考文献

Azevedo S. and Longini R.L. (1980) Abdominal-lead fetal electrocardiographic R-wave enhancement for heart rate determination. *IEEE Trans. Biomedical Engineering*, **27**(5), 255–60.

Bajpai, A.C., Calus I.M. and Fairley J.A. (1973) *Mathematics for Engineers and Scientists*, Volume 2. New York: Wiley.

Barlow J.S. and Rémond A. (1981) Eye movement artifact nulling in EEGs by multichannel on-line EOG subtraction. *Electroencephalography and Clinical Neurophysiology*, **52**, 418–23.

Bierman G.J. (1976) Measurement updating using the *U-D* factorization. *Automatica*, **12**, 375–82.

Bierman G.J. (1977) *Factorization Methods for Discrete Sequential Estimation*. New York: Academic Press.

Clark R.J., Ifeachor E.C., Van Eetvelt P.W.J. and Rogers G.M. (2000) Techniques for generating digital equalizer coefficients. *J. Audio Eng. Soc.*, **48**(4), April, 281–98.

Clarke D.W. (1981) Implementation of self-tuning controllers. In *Self-Tuning and Adaptive Control* (Harris C.J. and Billings S.A. (eds)), pp. 144–65. Stevenage, UK: Peter Peregrinus.

Cowan C.F.N. and Grant P.M. (1984) Adaptive processing – an overview. In *IEE Colloq. Adaptive Processing and Biomedical Applications*, October 1984, Paper 1.

Dattorro J. (1988) The implementation of recursive digital filters for high-fidelity audio. *J. Audio Eng. Soc.*, **36**(11), 851–78.

Favret A.G. (1968) Computer matched filter location of fetal R-waves. *Medical and Biological Engineering*, **6**, 467–75.

Flores I. (1963) *The Logic of Computer Arithmetic*. Englewood Cliffs NJ: Prentice-Hall.

Fortgens C. and De Bruin M.P. (1983) Removal of eye movement and ECG artifacts from the non-cephalic reference EEG. *Electroencephalography and Clinical Neurophysiology*, **56**, 90–6.

Girton D.G. and Kamiya A.J. (1973) A simple on-line technique for removing eye movement artifacts from the EEG. *Electroencephalography and Clinical Neurophysiology*, **34**, 212–8.

Goodman G.C. and Sin K.S. (1984) *Adaptive Filtering, Prediction and Control*. Englewood Cliffs NJ: Prentice-Hall.

Gotman J., Gloor P. and Ray W.F. (1975) A quantitative comparison of traditional reading of the EEG and interpretation of computer-extracted features in patients with supratentorial brain lesions. *Electroencephalography and Clinical Neurophysiology*, **38**, 623–39.

Greene K.R. (1987) The ECG waveform. In *Balliere's Clinical Obstetrics and Gynaecology* (Whittle M. (ed.)), Volume 1, pp. 131–55.

Hamer C.F., Ifeachor E.C. and Jervis B.W. (1985) Digital filtering of physiological signals with minimal distortion. *Medical and Biological Engineering and Computation*, **23**, 274–8.

Harris C.J. (1983) Brainwaves appear on T.V. in real-time. *Electronics*, February, 47–8.

IEEE (1985) IEEE Standard for Binary Floating Point Arithmetic. *SIGPLAN Notices*, **22**(2), 9–25.

Ifeachor E.C. (2001) *A Practical Guide for MATLAB and C Language Implementations of DSP Algorithms*. Harlow: Pearson Education.

Ifeachor E.C., Jervis B.W., Morris E.L., Allen E.M. and Hudson N.R. (1986) A new microcomputer-based online ocular artefact removal (OAR) system. *Proc. IEE*, **133**(5), 291–300.

Ifeachor E.C., Keith R.D.F., Westgate J. and Greene K.R. (1991) An expert system to assist in the management of labour. In *Proc. World Congress on Expert Systems* (Liebowitz J. (ed.)), Volume 4, pp. 2615–22. New York: Pergamon.

Jervis B.W., Allen E., Johnson T.E., Nichols M.J. and Hudson N.R. (1984) The application of pattern recognition techniques to the contingent negative variation for the differentiation of subject categories. *IEEE Trans. Biomedical Engineering*, **31**, 342–9.

Jervis B.W., Nichols M.J., Allen E., Hudson N.R. and Johnson T.E. (1985) The assessment of two methods for removing eye movement artefact from the EEG. *Electroencephalography and Clinical Neurophysiology*, **61**, 444–52.

Lindecrantz K.G., Lilja H. and Rosen K.G. (1988) New software QRS detector algorithm suitable for realtime applications with low signal to noise ratios. *J. Biomedical Engineering*, **10**, 280–3.

Motorola (1995) DSP56000 Digital Signal Processor Family Manual. Austin TX: Motorola.

Motorola (1996) DSP56302 Evaluation Module. Motorola Inc. *www1.motoroladsp.com/docs/docs.html*

Outram N.J., Ifeachor E.C., Van Eetvelt P.W.J. and Curnow J.S.H. (1995) Techniques for optimal enhancement and feature extraction of the fetal electrocardiogram. *IEE Proc.-Sci. Meas. Technol.*, **142**(6), 482–9.

Patterson D.A. and Hennessy J.L. (1990) *Computer Architecture: A Quantitative Approach*. San Mateo CA: Morgan Kaufmann.

Peterka V. (1975) A square root filter for real-time multivariate regression. *Kybernetika*, **11**, 53–67.

Quilter P.M., Macgillivray B.B. and Wadbrook D.G. (1977) The removal of eye movement artefact from EEG signals using correlation techniques. *IEE Conf. Publ.*, **159**, 93–100.

Rabiner L.R. and Gold B. (1975) *Theory and Application of Digital Signal Processing*. Englewood Cliffs NJ: Prentice-Hall.

Takeda H. and Hata S. (1985) Development of micro-computerized topographic EEG analyzer and its application to real time display. *Electroencephalography and Clinical Neurophysiology*, **61**, 98.

Texas Instruments (1986) *Digital Signal Processing Applications with the TMS320 Family: Theory, Algorithms and Implementations*. Texas Instruments.

Texas Instruments (1995) TMS320C54x evaluation module technical reference. Texas Instruments. *www.ti.com/sc/docs/psheets/abstract/apps/spru135.html*

Tomé A.M., Principe J.C. and Da Silva A.M. (1985) Micro analysis of spike and wave bursts in children's EEG. *Electroencephalography and Clinical Neurophysiology*, **61**, 113.

Weitek (1984) *High Speed Digital Arithmetic VLS Application Seminar Notes*. Sunnyvale CA: Weitek.

Widrow B., Glover J.R., McCool J.M., Kaunitz J., Williams C.S., Hearn R.H., Zeidler J.R., Dong E. and Goodlin R.C. (1975) Adaptive noise cancelling: principles and applications. *Proc. IEEE*, **63**, 1692–716.

Young P. (1974) Recursive approaches to time series analysis. *Bull. IMA*, **10**, 209–24.

参考书目

Clarke D.W. (1980) Some implementation considerations of self-tuning controllers. In *Numerical Techniques for Stochastic Systems* (Archetti F. and Cugiani M. (eds)), pp. 81–101. Amsterdam: North-Holland.

Clarke D.W., Cope S.N. and Gawthrop P.J. (1975) *Feasibility Study of the Application of Microprocessors to Self-tuning Controllers*. OUEL Report 1137/75.

Dattorro J. (1988) The implementation of recursive digital filters for high fidelity audio. *J. Audio Eng. Soc.*, **36**(11), 851–78.

Kay S.M. (1987) *Modern Spectrum Estimation*. Englewood Cliffs NJ: Prentice-Hall.

Marple S.L. Jr (1987) *Digital Spectral Analysis with Applications*. Englewood Cliffs, NJ: Prentice-Hall.

Motorola (1980) *16-bit Microprocessor User's Manual*. Austin TX: Motorola Semiconductor.

Otnes R.K. and Enochson L. (1978) *Applied Time Series Analysis*, Volume 1. New York: Wiley.

Rosen K.G. and Lindecrantz K.G. (1989) STAN, the Gothenburg model for fetal surveillance during labour by ST analysis of the fetal electrocardiogram. *Clinical Physiology and Physiological Measurement, Suppl. B*, **10**, 51–6.

Stanley W.D., Dougherty G.R. and Dougherty R. (1984) *Digital Signal Processing*, 2nd edn. Reston VA: Reston Publications.

Verleger R., Gasser T. and Möcks J. (1982) Correction of EOG artifacts in event-related potentials of the EEG: aspects of reliability and validity. *Psychophysiology*, **19**, 472–80.

附录

14A 修正的UD因子分解算法

- 步骤1：$\mathbf{v}=\mathbf{U}^{\mathrm{T}}(m)\mathbf{x}$
- 步骤2：$b_i=d_i(m)v_i,\quad i=1,\ldots,n$
- 步骤3：$\alpha_1=\gamma+b_1v_1$
- 步骤4：$d_1(m+1)=d_1(m)/\alpha_1$

对于$j=2,\ldots,n$，递归地计算14.9式～14.13式。

- 步骤5：$\alpha_j=\alpha_{j-1}+v_jb_j$
- 步骤6：$\rho_j=-v_j/\alpha_{j-1}$

对于$k=1,2,\ldots,j-1$，递归地计算14.11式。

- 步骤7：$U_{kj}(m+1)=U_{kj}(m)+b_k\rho_j$
- 步骤8：$b_k=b_k+b_jU_{kj}(m)$
- 步骤9：$d_j(m+1)=d_j(m)\alpha_{j-1}/\alpha_j\gamma$
- 步骤10：$G=b/\alpha_n\ (g_i=b_i/\alpha_n,\ i=1,\ldots,n)$

应该注意下列几点。

(1) 步骤10中的α_n是α_j（步骤5）第n次迭代后的数值。RLS算法14.9式中的α也是这样的。

(2) **D**的元素（即步骤4和步骤9中的d_i）可以沿着**U**的对角线存储，因为$U_{jj}=1$。另外，为了节省存储空间和易于编程，**U**的元素（包括**D**的）可以作为矢量存储，尽管下标(k,j)指明**U**是一个二维数组。